ektron

SDK

ektron CMS400

API Reference Documentation

Version 8.0

Copyright 2008 Ektron, Inc. All rights reserved www.ektron.com
EKTRON is a registered trademark of Ektron, Inc.

Table of Contents

Welcome 1

How Do I... 2

List of Developer Resources 4

Ektron Namespace 7

Server Controls API **7**

Ektron.Cms Namespace 7

Ektron.Cms.Controls Namespace 7

Ektron.Cms.Controls.CalendarProvider Namespace 11

Ektron.Cms.Controls.EktronUserControl Namespace 12

Classes 12

ActiveTopics Class 15

ActiveTopics Methods 16

ActiveTopics.Fill Method 16

ActiveTopics.ToString Method 16

ActiveTopics.Wrap Method 16

ActiveTopics Properties 17

ActiveTopics.BoardID Property 17

ActiveTopics.CacheInterval Property 17

ActiveTopics.DisplayMode Property 18

ActiveTopics.DynamicForumParameter Property 18

ActiveTopics.DynamicGroupParameter Property 18

ActiveTopics.DynamicThreadParameter Property 19

ActiveTopics.MaxNumber Property 19

ActiveTopics.URLPath Property 19

ActivityStream Class 20

ActivityStream Methods 20

ActivityStream.Fill Method 20

ActivityStream.LoadData Method 20

ActivityStream.RaiseCallbackEvent Method 21

ActivityStream Properties 21

ActivityStream.CacheInterval Property 21

ActivityStream.DefaultObjectID Property 22
ActivityStream.DisplayXslt Property 22
ActivityStream.DynamicObjectParameter Property 22
ActivityStream.EnablePaging Property 23
ActivityStream.MaxResults Property 23
ActivityStream.ObjectType Property 24
ActivityStream.ProfileParamName Property 24
ActivityStream.Stylesheet Property 24
ActivityStream.TemplateUserProfile Property 25
ActivityStream.XmlDoc Property 25
AnalyticsTracker Class 25
AnalyticsTracker Methods 26
AnalyticsTracker.Wrap Method 26
AnalyticsTracker Properties 26
AnalyticsTracker.DefaultContentID Property 26
AnalyticsTracker.DynamicParameter Property 27
AnalyticsTracker.EnableAnalytics Property 27
AssetControl Class 27
AssetControl Methods 28
AssetControl.Fill Method 28
AssetControl.ToString Method 28
AssetControl Properties 28
AssetControl.DefaultAssetID Property 29
AssetControl.DefaultFolderID Property 29
AssetControl.DynamicParameter Property 30
AssetControl.IsImage Property 30
AssetControl.OverrideExtension Property 30
AssetControl.TaxonomyID Property 31
AssetControl.UploadType Property 31
Blog Class 32
Blog Methods 32
Blog.Fill Method 32
Blog.ToString Method 32
Blog.Wrap Method 33
Blog Properties 33
Blog.ArchiveMode Property 33
Blog.BlogID Property 34
Blog.CacheInterval Property 34
Blog.DateToStart Property 35

Blog.DefaultUserID Property 35
Blog.DynamicParameter Property 35
Blog.DynamicUserParameter Property 36
Blog.EditorHeight Property 36
Blog.EditorWidth Property 36
Blog.JavascriptEditorHTMLMode Property 37
Blog.MaxResults Property 37
Blog.PostParameter Property 37
Blog.RecentPosts Property 38
Blog.ShowHeader Property 38
Blog.ShowRSS Property 38
BlogArchive Class 39
BlogArchive Methods 39
BlogArchive.Fill Method 39
BlogArchive.Wrap Method 39
BlogArchive Properties 40
BlogArchive.ArchiveMode Property 40
BlogArchive.BlogID Property 40
BlogArchive.CacheInterval Property 41
BlogArchive.DynamicParameter Property 41
BlogArchive.Text Property 41
BlogArchive.WrapTag Property 42
BlogCalendar Class 42
BlogCalendar Methods 42
BlogCalendar.Fill Method 42
BlogCalendar.Wrap Method 42
BlogCalendar Properties 43
BlogCalendar.BlogID Property 43
BlogCalendar.CacheInterval Property 43
BlogCalendar.Text Property 44
BlogCalendar.WrapTag Property 44
BlogCategories Class 44
BlogCategories Methods 45
BlogCategories.Fill Method 45
BlogCategories.Wrap Method 45
BlogCategories Properties 45
BlogCategories.BlogID Property 45
BlogCategories.CacheInterval Property 46
BlogCategories.Text Property 46

BlogCategories.WrapTag Property 46
BlogEntries Class 47
BlogEntries Methods 47
BlogEntries.Fill Method 47
BlogEntries.GetRssFeed Method 47
BlogEntries.GetView Method 48
BlogEntries.GetViewNames Method 48
BlogEntries.ToString Method 48
BlogEntries.Wrap Method 48
BlogEntries Properties 49
BlogEntries.BlogID Property 49
BlogEntries.BlogPostParameter Property 50
BlogEntries.CacheInterval Property 50
BlogEntries.DateToStart Property 50
BlogEntries.DefaultUserID Property 51
BlogEntries.DisplayXslt Property 51
BlogEntries.DynamicParameter Property 51
BlogEntries.DynamicUserParameter Property 52
BlogEntries.EditorHeight Property 52
BlogEntries.EditorWidth Property 52
BlogEntries.JavascriptEditorHTMLMode Property 53
BlogEntries.MaxResults Property 53
BlogEntries.ShowHeader Property 54
BlogEntries.XmlDoc Property 54
BlogPost Class 54
BlogPost Methods 54
BlogPost.Wrap Method 55
BlogPost Properties 55
BlogPost.DefaultContentID Property 55
BlogPost.ShowType Property 56
BlogPost.XmlDoc Property 56
BlogRecentPosts Class 56
BlogRecentPosts Methods 57
BlogRecentPosts.Fill Method 57
BlogRecentPosts.Wrap Method 57
BlogRecentPosts Properties 57
BlogRecentPosts.BlogID Property 57
BlogRecentPosts.CacheInterval Property 58
BlogRecentPosts.NumberofPosts Property 58

BlogRecentPosts.Text Property 59
BlogRecentPosts.WrapTag Property 59
BlogRoll Class 59
BlogRoll Methods 59
BlogRoll.Fill Method 59
BlogRoll.Wrap Method 60
BlogRoll Properties 60
BlogRoll.BlogID Property 60
BlogRoll.CacheInterval Property 61
BlogRoll.Text Property 61
BlogRoll.WrapTag Property 61
BlogRSS Class 61
BlogRSS Methods 62
BlogRSS.Fill Method 62
BlogRSS Properties 62
BlogRSS.BlogID Property 62
BlogRSS.CacheInterval Property 63
BlogRSS.Text Property 63
BlogRSS.WrapTag Property 63
BreadCrumb Class 63
BreadCrumb Methods 64
BreadCrumb.Fill Method 64
BreadCrumb.GenerateTrail_HTML Method 64
ResetTrail Method 64
BreadCrumb Properties 66
BreadCrumb.CurrentPageIndicator Property 66
BreadCrumb.DefaultContentID Property 67
BreadCrumb.DisplayStyle Property 67
BreadCrumb.DisplayTitle Property 67
BreadCrumb.DynamicParameter Property 67
BreadCrumb.IconAlt Property 68
BreadCrumb.IconPath Property 68
BreadCrumb.LinkLastItem Property 68
BreadCrumb.LinkTarget Property 68
BreadCrumb.MaxItems Property 69
BreadCrumb.MetadataName Property 69
BreadCrumb.Mode Property 69
BreadCrumb.separator Property 70
BreadCrumbItem Class 70

BreadCrumbItem Constructor 70
BreadCrumbItem.BreadCrumbItem Constructor () 70
BreadCrumbItem.BreadCrumbItem Constructor (string, string, string, string, string) 71
BreadCrumbItem Methods 71
BreadCrumbItem.Equals Method 71
BreadCrumbItem.GetHashCode Method 71
BreadCrumbItem.ToString Method 72
BreadCrumbItem Properties 72
BreadCrumbItem.CaseSensitiveUrlComparison Property 72
BreadCrumbItem.DisplayTitle Property 72
BreadCrumbItem.IconAlt Property 72
BreadCrumbItem.IconPath Property 73
BreadCrumbItem.RawUrl Property 73
BreadCrumbItem.Target Property 73
BusinessRules Class 73
BusinessRules Methods 74
BusinessRules.Fill Method 74
BusinessRules.Wrap Method 74
BusinessRules Properties 74
BusinessRules.Parameters Property 75
BusinessRules.Result Property 75
BusinessRules.Text Property 76
BusinessRules.WrapTag Property 76
Calendar Class 76
Calendar Methods 76
Calendar.Fill Method 77
Calendar.Wrap Method 77
Calendar Properties 77
Calendar.DefaultCalendarID Property 78
Calendar.DisplayType Property 78
Calendar.DynamicParameter Property 78
Calendar.EndDate Property 79
Calendar.HighlightEventID Property 79
Calendar.HighlightEventTypeID Property 79
Calendar.ShowXML Property 80
Calendar.StartDate Property 80
Calendar.Text Property 80
Cart Class 81
Cart Properties 81

Cart.DisplayXslt Property 82
Cart.DynamicProductParameter Property 82
Cart.EnableCoupons Property 82
Cart.LoadingImage Property 83
Cart.Settings Property 83
Cart.Stylesheet Property 83
Cart.TemplateCheckout Property 84
Cart.TemplateProduct Property 84
Cart.TemplateShopping Property 85
Cart.UserControlUrl Property 85
Checkout Class 85
Checkout Enumerations 86
Ektron.Cms.Controls.Checkout.ControlPhase Enumeration 86
Checkout Methods 86
Checkout.Fill Method 86
Checkout.ToString Method 86
Checkout.Wrap Method 87
Checkout Properties 87
Checkout.Cart Property 88
Checkout.CurrentPhase Property 88
Checkout.Customer Property 88
Checkout.DefaultCountryId Property 89
Checkout.DisplayXslt Property 89
Checkout.DynamicPhaseParameter Property 89
Checkout.FriendlyErrorMessage Property 90
Checkout.IsSSLRequired Property 90
Checkout.LoadingImage Property 91
Checkout.PaymentMethod Property 91
Checkout.ShowErrorDetails Property 91
Checkout.Stylesheet Property 92
Checkout.TemplateCart Property 92
Checkout.TemplateOrderHistory Property 92
Checkout.TemplateRecoverPassword Property 93
Checkout.TemplateShopping Property 93
Checkout.UseMsAjax Property 94
Checkout.ValidatePhase Property 94
Checkout Delegates 94
Checkout.PhaseChangeEventHandler Delegate 94
Checkout.PostLoginEventHandler Delegate 95

Checkout Events 95
Checkout.PhaseChange Event 95
Checkout.PostLogin Event 95
Collection Class 95
Collection Methods 96
Collection.CopyTo Method 96
Collection.Fill Method 97
Collection.GetEnumerator Method 97
Collection.GetRssFeed Method 97
Collection.GetView Method 97
Collection.GetViewNames Method 98
Collection.MoveNext Method 98
Collection.Reset Method 98
Collection.this Indexer 98
Collection.ToString Method 98
Collection.Wrap Method 98
Collection Properties 99
Collection.AddText Property 100
Collection.CacheInterval Property 100
Collection.CollectionDescription Property 100
Collection.CollectionTitle Property 101
Collection.ContentParameter Property 101
Collection.Count Property 101
Collection.Current Property 101
Collection.DefaultCollectionID Property 102
Collection.DynamicParameter Property 102
Collection.EkItems Property 102
Collection.EnablePaging Property 103
Collection.GetHtml Property 103
Collection.IncludeIcons Property 103
Collection.IsSynchronized Property 104
Collection.LinkTarget Property 104
Collection.MarkupLanguage Property 104
Collection.MaxResults Property 104
Collection.Random Property 105
Collection.SelTaxonomyID Property 105
Collection.SuppressWrapperTags Property 105
Collection.SyncRoot Property 106
Collection.XmlDoc Property 106

CommunityDocuments Class 106
CommunityDocuments Methods 106
CommunityDocuments.Fill Method 107
CommunityDocuments Properties 107
CommunityDocuments.AddItemFolderId Property 107
CommunityDocuments.DefaultObjectID Property 108
CommunityDocuments.DynamicContentBoxHeight Property 108
CommunityDocuments.DynamicContentBoxWidth Property 108
CommunityDocuments.DynamicContentTemplate Property 109
CommunityDocuments.EnableAjax Property 109
CommunityDocuments.EnableSearch Property 109
CommunityDocuments.MemberMenuActive Property 109
CommunityDocuments.ObjectType Property 110
CommunityDocuments.Recursive Property 110
CommunityDocuments.ShowAllChecked Property 110
CommunityDocuments.Stylesheet Property 110
CommunityDocuments.SuppressWrapperTags Property 111
CommunityDocuments.TaxonomyDepth Property 111
CommunityDocuments.TaxonomyId Property 111
CommunityGroupBrowser Class 111
CommunityGroupBrowser Methods 112
CommunityGroupBrowser.Fill Method 112
CommunityGroupBrowser.ToString Method 112
CommunityGroupBrowser.Wrap Method 112
CommunityGroupBrowser Properties 113
CommunityGroupBrowser.AvatarHeight Property 113
CommunityGroupBrowser.AvatarWidth Property 113
CommunityGroupBrowser.CacheInterval Property 114
CommunityGroupBrowser.DisplayXslt Property 114
CommunityGroupBrowser.Link Property 115
CommunityGroupBrowser.LinkTarget Property 115
CommunityGroupBrowser.MarkupLanguage Property 115
CommunityGroupBrowser.MaxResults Property 115
CommunityGroupBrowser.Stylesheet Property 116
CommunityGroupBrowser.TagTemplate Property 116
CommunityGroupBrowser.TagTemplateTarget Property 116
CommunityGroupBrowser.TaxonomyId Property 116
CommunityGroupBrowser.XmlDoc Property 117
CommunityGroupList Class 117

CommunityGroupList Methods 117
CommunityGroupList.Fill Method 118
CommunityGroupList.ToString Method 118
CommunityGroupList.Wrap Method 118
CommunityGroupList Properties 118
CommunityGroupList.AllowGroupCreation Property 119
CommunityGroupList.AvatarHeight Property 119
CommunityGroupList.AvatarWidth Property 119
CommunityGroupList.CacheInterval Property 119
CommunityGroupList.DefaultUserID Property 120
CommunityGroupList.DisplayMode Property 120
CommunityGroupList.DisplayXslt Property 120
CommunityGroupList.DynamicUserParameter Property 121
CommunityGroupList.LanguageId Property 121
CommunityGroupList.Link Property 122
CommunityGroupList.LinkTarget Property 122
CommunityGroupList.MarkupLanguage Property 122
CommunityGroupList.MaxResults Property 122
CommunityGroupList.ShowMySortingOptions Property 123
CommunityGroupList.ShowSortingOptions Property 123
CommunityGroupList.Stylesheet Property 123
CommunityGroupList.TagTemplate Property 123
CommunityGroupList.TagTemplateTarget Property 124
CommunityGroupList.XmlDoc Property 124
CommunityGroupMembers Class 124
CommunityGroupMembers Methods 125
CommunityGroupMembers.Fill Method 125
CommunityGroupMembers.ToString Method 125
CommunityGroupMembers.Wrap Method 125
CommunityGroupMembers Properties 125
CommunityGroupMembers.AvatarHeight Property 126
CommunityGroupMembers.AvatarWidth Property 126
CommunityGroupMembers.CacheInterval Property 126
CommunityGroupMembers.DefaultCommunityGroupID Property 127
CommunityGroupMembers.DisplayMode Property 127
CommunityGroupMembers.DisplayXslt Property 127
CommunityGroupMembers.DynamicParameter Property 128
CommunityGroupMembers.Link Property 128
CommunityGroupMembers.LinkTarget Property 128

CommunityGroupMembers.MarkupLanguage Property 129
CommunityGroupMembers.MaxResults Property 129
CommunityGroupMembers.Stylesheet Property 129
CommunityGroupMembers.XmlDoc Property 130
CommunityGroupProfile Class 130
CommunityGroupProfile Methods 130
CommunityGroupProfile.Fill Method 130
CommunityGroupProfile.ToString Method 130
CommunityGroupProfile.Wrap Method 131
CommunityGroupProfile Properties 131
CommunityGroupProfile.DefaultGroupID Property 131
CommunityGroupProfile.DisplayXslt Property 132
CommunityGroupProfile.DynamicParameter Property 132
CommunityGroupProfile.EkItem Property 132
CommunityGroupProfile.GroupAdminLink Property 133
CommunityGroupProfile.GroupAdminLinkTarget Property 133
CommunityGroupProfile.Stylesheet Property 133
CommunityGroupProfile.TagTemplate Property 133
CommunityGroupProfile.TagTemplateTarget Property 134
CommunityGroupProfile.Text Property 134
CommunityGroupProfile.XmlDoc Property 134
CommunitySearch Class 135
CommunitySearch Methods 135
CommunitySearch.Fill Method 135
CommunitySearch.ToString Method 135
CommunitySearch.Wrap Method 136
CommunitySearch Properties 136
CommunitySearch.CacheInterval Property 137
CommunitySearch.CustomPropertyID Property 138
CommunitySearch.DisplayXslt Property 138
CommunitySearch.EnableFilteringResults Property 139
CommunitySearch.EnableGroupResults Property 139
CommunitySearch.EnableMap Property 139
CommunitySearch.EnableMouseOutHidesInfo Property 140
CommunitySearch.EnableMouseOver Property 140
CommunitySearch.EnableUnauthorizedSearching Property 140
CommunitySearch.EnableUserResults Property 141
CommunitySearch.FriendsOnly Property 141
CommunitySearch.GroupTaxonomyID Property 141

CommunitySearch.InfoBoxShowsEmail Property 142
CommunitySearch.LoadingImage Property 142
CommunitySearch.MapControl Property 142
CommunitySearch.MarkupLanguage Property 143
CommunitySearch.MaxTagCount Property 143
CommunitySearch.MembersOnly Property 143
CommunitySearch.PageSize Property 144
CommunitySearch.ResultProfile Property 144
CommunitySearch.SearchPrivate Property 144
CommunitySearch.ShowNameEmail Property 145
CommunitySearch.StartingTab Property 145
CommunitySearch.Stylesheet Property 146
CommunitySearch.TemplateGroupProfile Property 146
CommunitySearch.TemplateParamName Property 146
CommunitySearch.TemplateTarget Property 147
CommunitySearch.TemplateUserProfile Property 147
CommunitySearch.UseMsAjax Property 147
CommunitySearch.UserTaxonomyID Property 148
CommunitySearch.XmlDoc Property 148
CommunitySearch.XsltDoc Property 148
ContentBlock Class 148
ContentBlock Methods 149
ContentBlock.CopyTo Method 149
ContentBlock.Fill Method 150
ContentBlock.GetEnumerator Method 150
ContentBlock.MoveNext Method 150
ContentBlock.Reset Method 150
ContentBlock.this Indexer 150
ContentBlock.ToString Method 150
ContentBlock.Wrap Method 151
ContentBlock Properties 151
ContentBlock.CacheInterval Property 151
ContentBlock.Count Property 152
ContentBlock.Current Property 152
ContentBlock.DefaultContentID Property 152
ContentBlock.DisplayXslt Property 153
ContentBlock.DynamicParameter Property 153
ContentBlock.EkItem Property 153
ContentBlock.IsSynchronized Property 154

ContentBlock.OverrideXslt Property 154
ContentBlock.StartingFolderID Property 154
ContentBlock.SyncRoot Property 155
ContentFlagging Class 155
ContentFlagging Methods 155
ContentFlagging.Fill Method 155
ContentFlagging.ToString Method 155
ContentFlagging.Wrap Method 156
ContentFlagging Properties 156
ContentFlagging.DefaultContentID Property 156
ContentFlagging.DisplayXslt Property 157
ContentFlagging.DynamicParameter Property 157
ContentFlagging.FlagImage Property 157
ContentFlagging.FlagText Property 158
ContentFlagging.Stylesheet Property 158
ContentFlagging.XmlDoc Property 159
ContentList Class 159
ContentList Methods 159
ContentList.CopyTo Method 160
ContentList.Fill Method 160
ContentList.GetEnumerator Method 160
ContentList.GetRssFeed Method 160
ContentList.MoveNext Method 160
ContentList.Reset Method 161
ContentList.this Indexer 161
ContentList.ToString Method 161
ContentList.Wrap Method 161
ContentList Properties 162
ContentList.CacheInterval Property 163
ContentList.ContentIds Property 163
ContentList.Count Property 164
ContentList.Current Property 164
ContentList.DefaultContentID Property 164
ContentList.Direction Property 164
ContentList.DynamicParameter Property 165
ContentList.EkItems Property 165
ContentList.GetHtml Property 165
ContentList.IncludeIcons Property 166
ContentList.IsSynchronized Property 166

ContentList.LinkTarget Property 166
ContentList.MarkupLanguage Property 167
ContentList.MetaTag Property 167
ContentList.OrderByDirection Property 167
ContentList.OrderKey Property 168
ContentList.Random Property 168
ContentList.SuppressWrapperTags Property 168
ContentList.SyncRoot Property 169
ContentList.XmlDoc Property 169
ContentRating Class 169
ContentRating Methods 169
ContentRating.Fill Method 170
ContentRating.ToString Method 170
ContentRating.Wrap Method 170
ContentRating Properties 170
ContentRating.AlreadyRatedMessage Property 171
ContentRating.BadLabel Property 171
ContentRating.CharsRemainingLabel Property 172
ContentRating.CompleteGraphDisplay Property 172
ContentRating.CookiesRequiredMessage Property 172
ContentRating.DefaultContentID Property 173
ContentRating.DynamicParameter Property 173
ContentRating.FeedbackHeader Property 173
ContentRating.GoodLabel Property 174
ContentRating.GraphBarColor Property 174
ContentRating.InitialGraphDisplay Property 174
ContentRating.Padding Property 175
ContentRating.RatingLevelLabel Property 175
ContentRating.ThankYouMessage Property 176
ContentRating.TotalRatingsLabel Property 176
ContentRating.UserCommentsHeader Property 176
ContentRating.ValidationMessage Property 177
ContentReview Class 177
ContentReview Methods 177
ContentReview.Fill Method 178
ContentReview.RaiseCallbackEvent Method 178
ContentReview.ToString Method 178
ContentReview.Wrap Method 178
ContentReview Properties 179

ContentReview.DefaultContentID Property 179
ContentReview.DisplayXslt Property 180
ContentReview.DynamicParameter Property 180
ContentReview.DynamicUserParameter Property 181
ContentReview.GetReviews Property 181
ContentReview.MaxReviews Property 181
ContentReview.Moderate Property 182
ContentReview.RatingsMinimum Property 182
ContentReview.UserID Property 182
ContentReview.XmlDoc Property 183
CurrencySelect Class 183
CurrencySelect Methods 183
CurrencySelect.Fill Method 184
CurrencySelect.ToString Method 184
CurrencySelect.Wrap Method 184
CurrencySelect Properties 184
CurrencySelect.CacheInterval Property 184
CurrencySelect.DisplayXslt Property 185
CurrencySelect.EkItem Property 186
CurrencySelect.Stylesheet Property 186
DesignTimeDiagnostic Class 186
DesignTimeDiagnostic Methods 187
DesignTimeDiagnostic.Fill Method 187
DesignTimeDiagnostic Properties 187
DesignTimeDiagnostic.Authenticated Property 187
DesignTimeDiagnostic.Hide Property 187
DesignTimeDiagnostic.Language Property 188
DesignTimeDiagnostic.SuppressWrapperTags Property 188
DesignTimeDiagnostic.WrapTag Property 188
DhtmlMenu Class 188
DhtmlMenu Methods 189
DhtmlMenu.Fill Method 189
DhtmlMenu.Wrap Method 189
DhtmlMenu Properties 189
DhtmlMenu.AccessKey Property 190
DhtmlMenu.Align Property 190
DhtmlMenu.BackColor Property 191
DhtmlMenu.BorderColor Property 191
DhtmlMenu.BorderStyle Property 191

DhtmlMenu.BorderWidth Property 191
DhtmlMenu.CssClass Property 192
DhtmlMenu.DisplayXslt Property 192
DhtmlMenu.Enabled Property 192
DhtmlMenu.EnableIE6FormsFix Property 192
DhtmlMenu.Height Property 193
DhtmlMenu.HorizontalDisplay Property 193
DhtmlMenu.MenuHoverOverColor Property 193
DhtmlMenu.MenuStyle Property 194
DhtmlMenu.Stylesheet Property 194
DhtmlMenu.SubMenuBackColor Property 194
DhtmlMenu.SubMenuHoverOverColor Property 194
DhtmlMenu.SubMenuItemHeight Property 195
DhtmlMenu.SubMenuItemWidth Property 195
DhtmlMenu.SubMenuTopOffset Property 195
DhtmlMenu.TabIndex Property 196
DhtmlMenu.ToolTip Property 196
DhtmlMenu.Width Property 196
Directory Class 196
Directory Methods 197
Directory.CopyTo Method 197
Directory.Fill Method 197
Directory.GetEnumerator Method 198
Directory.GetView Method 198
Directory.GetViewNames Method 198
Directory.MoveNext Method 198
Directory.Reset Method 199
Directory.this Indexer 199
Directory.ToString Method 199
Directory.Wrap Method 199
Directory Properties 200
Directory.AddItemFolderId Property 201
Directory.CacheInterval Property 201
Directory.Count Property 202
Directory.Current Property 202
Directory.DisplayXslt Property 202
Directory.DynamicParameter Property 202
Directory.EkItems Property 203
Directory.EnableAjax Property 203

Directory.EnablePaging Property 203
Directory.EnableSearch Property 204
Directory.IncludeIcons Property 204
Directory.IsSynchronized Property 204
Directory.ItemSortOrder Property 204
Directory.LinkTarget Property 205
Directory.MarkupLanguage Property 205
Directory.MaxResults Property 205
Directory.Recursive Property 206
Directory.ShowAllChecked Property 206
Directory.ShowCount Property 206
Directory.ShowRoot Property 207
Directory.SortDirection Property 207
Directory.SuppressWrapperTags Property 207
Directory.SyncRoot Property 208
Directory.TaxonomyCols Property 208
Directory.TaxonomyDepth Property 208
Directory.TaxonomyDescription Property 208
Directory.TaxonomyId Property 209
Directory.TaxonomyItemCols Property 209
Directory.TaxonomyParamId Property 209
Directory.TaxonomyPath Property 209
Directory.TaxonomyTitle Property 210
Directory.TaxonomyTreeData Property 210
Directory.TaxonomyXml Property 210
Directory.XmlDoc Property 210
EktronCatalogPart Class 210
EktronCatalogPart Properties 211
EktronCatalogPart.BlogID Property 211
EktronCatalogPart.BoardID Property 212
EktronCatalogPart.CalendarDynamicParameter Property 212
EktronCatalogPart.CollectionDynamicParameter Property 212
EktronCatalogPart.ContentBlockDynamicParameter Property 213
EktronCatalogPart.DefaultCalendarID Property 213
EktronCatalogPart.DefaultCollectionID Property 213
EktronCatalogPart.DefaultContentID Property 213
EktronCatalogPart.DefaultFormID Property 214
EktronCatalogPart.EnableBlogControl Property 214
EktronCatalogPart.EnableCalendarControl Property 214

EktronCatalogPart.EnableCollectionControl Property 215
EktronCatalogPart.EnableContentBlockControl Property 215
EktronCatalogPart.EnableDiscussionBoardsControl Property 215
EktronCatalogPart.EnableFormBlockControl Property 215
EktronCatalogPart.EnableListSummaryControl Property 216
EktronCatalogPart.EnableThreadedDiscussionsControl Property 216
EktronCatalogPart.FolderID Property 216
EktronCatalogPart.FormBlockDynamicParameter Property 216
EktronCatalogPart.Title Property 217
EktronCommunityCatalogPart Class 217
EktronCommunityCatalogPart Properties 217
EktronCommunityCatalogPart.EnableBlogs Property 218
EktronCommunityCatalogPart.EnableDiscussionBoards Property 218
EktronCommunityCatalogPart.EnableForums Property 218
EktronCommunityCatalogPart.Title Property 219
EktronWebPartZone Class 219
EktronWebPartZone Properties 219
EktronWebPartZone.CmsSitePath Property 219
EkWebControl Class 220
EkWebControl.EkWebControl Constructor 220
EkWebControl Methods 220
EkWebControl.Fill Method 221
EkWebControl.GetEnumDescription Method 221
EkWebControl.GetFilenameFromPath Method 221
GetMessage Method 222
EkWebControl.LocalizeString Method 222
EkWebControl.RedirectFromLoginPage Method 223
EkWebControl.RedirectToLoginPage Method 223
EkWebControl.RegisterAllDmsMenuCssAndScripts Method 223
EkWebControl.RegisterDmsMenuCSS Method 223
EkWebControl.RegisterDmsMenuScript Method 224
EkWebControl.RegisterJQueryScript Method 224
EkWebControl.RegisterMembershipTabScript Method 224
EkWebControl.RegisterThickBoxCSS Method 224
EkWebControl.RegisterThickBoxScript Method 224
EkWebControl.ToString Method 224
EkWebControl.Wrap Method 225
EkWebControl Properties 225
EkWebControl.Authenticated Property 225

EkWebControl.CmsSitePath Property 226
EkWebControl.Error Property 226
EkWebControl.Hide Property 226
EkWebControl.IsLoggedIn Property 227
EkWebControl.Language Property 227
EkWebControl.LanguageID Property 227
EkWebControl.loggedInUserID Property 227
EkWebControl.loggedInUserName Property 228
EkWebControl.LogLevel Property 228
EkWebControl.SuppressWrapperTags Property 228
EkWebControl.Text Property 228
EkWebControl.WrapTag Property 229
EkWebPart Class 229
EkWebPart Methods 230
EkWebPart.CreateEditorParts Method 230
EkWebPart Properties 230
EkWebPart.CatalogIconImageUrl Property 230
EkWebPart.Description Property 230
EkWebPart.Subtitle Property 231
EkWebPart.Title Property 231
EkWebPart.TitleIconImageUrl Property 231
EkWebPart.TitleUrl Property 231
EkWebPart.WebBrowsableObject Property 231
EkXsltBase Class 232
EkXsltBase.EkXsltBase Constructor 232
EkXsltBase Properties 232
EkXsltBase.CustomXml Property 232
EkXsltBase.DisplayXslt Property 233
EkXsltBase.EnableErrorLogging Property 233
EkXsltBase.XmlDoc Property 233
EkXsltBase.XsltDoc Property 234
ExplorerDragDrop Class 234
ExplorerDragDrop Properties 234
ExplorerDragDrop.AssetID Property 234
ExplorerDragDrop.ContentLanguage Property 235
ExplorerDragDrop.FolderID Property 235
ExplorerDragDrop.IsImage Property 235
ExplorerDragDrop.OverrideExtension Property 236
ExplorerDragDrop.TaxonomyId Property 236

Favorites Class 236
Favorites Methods 237
Favorites.Fill Method 237
Favorites.ToString Method 237
Favorites.Wrap Method 237
Favorites Properties 237
Favorites.CacheInterval Property 238
Favorites.DefaultUserID Property 239
Favorites.DisplayMode Property 239
Favorites.DisplayXslt Property 239
Favorites.DynamicParameter Property 240
Favorites.FavoritesTitle Property 240
Favorites.FavoritesUriTitle Property 240
Favorites.Link Property 240
Favorites.LinkTarget Property 241
Favorites.MarkupLanguage Property 241
Favorites.MaxResults Property 241
Favorites.Stylesheet Property 242
Favorites.XmlDoc Property 242
FlexMenu Class 242
FlexMenu Methods 243
FlexMenu.Fill Method 243
FlexMenu.SetDirty Method 243
FlexMenu Properties 243
FlexMenu.AutoCollapseBranches Property 244
FlexMenu.CacheInterval Property 244
FlexMenu.CacheIntervalAuthenticated Property 245
FlexMenu.DisplayXslt Property 245
FlexMenu.EnableAjax Property 245
FlexMenu.EnableMouseOverPopUp Property 246
FlexMenu.EnableSmartOpen Property 246
FlexMenu.EnableSubmenuLinkPathMatching Property 246
FlexMenu.EnableSubmenuTemplateMatching Property 247
FlexMenu.LogInfo Property 247
FlexMenu.MasterControlId Property 247
FlexMenu.MenuDepth Property 248
FlexMenu.StartCollapsed Property 248
FlexMenu.StartLevel Property 248
FlexMenu.Stylesheet Property 249

FlexMenu.UseAliasRealId Property 249
FlexMenu.XmlDoc Property 249
FlexMenu.XsltDoc Property 250
FolderBreadcrumb Class 250
FolderBreadcrumb Methods 250
FolderBreadcrumb.Fill Method 250
FolderBreadcrumb Properties 251
FolderBreadcrumb.DefaultContentID Property 251
FolderBreadcrumb.DefaultFolderID Property 251
FolderBreadcrumb.DisplayStyle Property 252
FolderBreadcrumb.DisplayXslt Property 252
FolderBreadcrumb.DynamicParameter Property 253
FolderBreadcrumb.LinkTarget Property 253
FolderBreadcrumb.Mode Property 253
FolderBreadcrumb.separator Property 254
FolderBreadcrumb.XmlDoc Property 254
FormBlock Class 254
FormBlock Methods 254
FormBlock.CopyTo Method 255
FormBlock.Fill Method 255
FormBlock.GetEnumerator Method 255
FormBlock.GetView Method 255
FormBlock.GetViewNames Method 256
FormBlock.MoveNext Method 256
FormBlock.Reset Method 256
FormBlock.this Indexer 256
FormBlock.ToString Method 256
FormBlock.Wrap Method 257
FormBlock Properties 257
FormBlock.AddValidation Property 257
FormBlock.CacheInterval Property 258
FormBlock.Count Property 258
FormBlock.Current Property 258
FormBlock.DefaultFormID Property 259
FormBlock.DynamicParameter Property 259
FormBlock.EkItem Property 259
FormBlock.Fields Property 260
FormBlock.FormName Property 260
FormBlock.IncludeTags Property 260

FormBlock.IsSynchronized Property 261
FormBlock.StartingFolderID Property 261
FormBlock.SyncRoot Property 261
FormFieldItem Class 261
FormFieldItem Properties 262
FormFieldItem.Basetype Property 262
FormFieldItem.ContentType Property 262
FormFieldItem.DatalistItems Property 263
FormFieldItem.Datatype Property 263
FormFieldItem.DisplayName Property 263
FormFieldItem.FieldName Property 263
FormFieldItem.Language Property 264
FormFieldItem.Value Property 264
Forum Class 264
Forum Methods 265
Forum.Fill Method 265
Forum.GetCallbackResult Method 265
Forum.RaiseCallbackEvent Method 265
Forum.ToString Method 266
Forum.Wrap Method 266
Forum Properties 266
Forum.AllowAnonymousPost Property 268
Forum.BoardID Property 268
Forum.CacheInterval Property 269
Forum.CustomOrderBy Property 269
Forum.DefaultObjectID Property 269
Forum.DynamicForumParameter Property 270
Forum.DynamicGroupParameter Property 270
Forum.DynamicObjectParameter Property 270
Forum.DynamicParameter Property 271
Forum.DynamicThreadParameter Property 271
Forum.EditorCSS Property 271
Forum.EditorToolbar Property 271
Forum.EnableForumQuickSelect Property 272
Forum.EnableThreadRating Property 272
Forum.ExcludeList Property 273
Forum.FilterXslt Property 273
Forum.HideUserProfile Property 273
Forum.HighlightedUserRole Property 274

Forum.JavascriptEditorHeight Property 274
Forum.JavascriptEditorHTMLMode Property 275
Forum.JavascriptEditorToolbar Property 275
Forum.JavascriptEditorWidth Property 276
Forum.ObjectType Property 276
Forum.OrderBy Property 276
Forum.OrderDirection Property 277
Forum.ProfileLink Property 277
Forum.ProfileLinkTarget Property 278
Forum.ShowCategories Property 278
Forum.TaxCategories Property 278
Forum.TaxonomyCols Property 278
Forum.TaxonomyMarkupLanguage Property 279
Forum.Theme Property 279
Friends Class 279
Friends Methods 280
Friends.Fill Method 280
Friends.ToString Method 280
Friends.Wrap Method 280
Friends Properties 280
Friends.AvatarHeight Property 281
Friends.AvatarWidth Property 282
Friends.CacheInterval Property 282
Friends.DefaultUserID Property 282
Friends.DisplayMode Property 283
Friends.DisplayXslt Property 283
Friends.DynamicParameter Property 283
Friends.FriendLabel Property 284
Friends.FriendsLabel Property 284
Friends.Link Property 284
Friends.LinkTarget Property 285
Friends.MarkupLanguage Property 285
Friends.MaxResults Property 285
Friends.ShowSortingOptions Property 285
Friends.Stylesheet Property 286
Friends.XmlDoc Property 286
HtmlEditDisplay Class 286
HtmlEditDisplay Methods 287
HtmlEditDisplay.BaseJavaScript Method 287

HtmlEditDisplay.BeginJavaScript Method 288
HtmlEditDisplay.EndJavaScript Method 288
HtmlEditDisplay.MakeFieldNameFromId Method 288
HtmlEditDisplay.RenderCMSFields Method 288
HtmlEditDisplay.SelectedJavaScript Method 289
HtmlEditor Class 289
HtmlEditor.HtmlEditor Constructor 290
HtmlEditor Methods 290
HtmlEditor.ExecuteContentAction Method 291
HtmlEditor.ToString Method 291
HtmlEditor.WorkareaMode Method 291
HtmlEditor Properties 292
HtmlEditor.Archived Property 293
HtmlEditor.Authenticated Property 294
HtmlEditor.CmsSitePath Property 294
HtmlEditor.ContentAction Property 295
HtmlEditor.ContentID Property 295
HtmlEditor.ContentTitle Property 296
HtmlEditor.ContentType Property 296
HtmlEditor.EditorReadyCall Property 297
HtmlEditor.EnableUpload Property 297
HtmlEditor.IncludeContentInfo Property 298
HtmlEditor.IsLoggedIn Property 298
HtmlEditor.Language Property 298
HtmlEditor.loggedInUserID Property 299
HtmlEditor.loggedInUserName Property 299
HtmlEditor.Path Property 300
HtmlEditor.PreferredType Property 300
HtmlEditor.SafeEncode Property 301
HtmlEditor.ShowCMSToolbar Property 301
HtmlEditor.ShowTitle Property 302
HtmlEditor.StyleSheet Property 302
HtmlEditor.Text Property 303
HtmlEditor.ToolbarLevel Property 303
HtmlEditor.ToolbarResetCall Property 304
HtmlEditor.TranslationFile Property 304
ImageControl Class 305
ImageControl Methods 305
ImageControl.Fill Method 305

ImageControl.Wrap Method 306
ImageControl Properties 306
ImageControl.DefaultImageID Property 306
ImageControl.DynamicParameter Property 306
ImageControl.FolderID Property 307
ImageControl.TaxonomyID Property 307
ImageControl.Title Property 308
IndexSearch Class 308
IndexSearch.IndexSearch Constructor 308
IndexSearch Structures 309
Ektron.Cms.Controls.IndexSearch.SearchPram Structure 309
IndexSearch Methods 309
AddParm Method 309
IndexSearch.CopyTo Method 310
IndexSearch.Fill Method 310
IndexSearch.GetEnumerator Method 311
IndexSearch.GetSearchParmDataType Method 311
IndexSearch.GetSearchParmSearchType Method 311
IndexSearch.GetXPathValues Method 311
IndexSearch.MoveNext Method 312
IndexSearch.Reset Method 312
Search Method 312
IndexSearch.this Indexer 313
IndexSearch.Wrap Method 313
IndexSearch Properties 313
IndexSearch.ButtonImgSrc Property 314
IndexSearch.ButtonText Property 315
IndexSearch.ContentParameter Property 315
IndexSearch.Count Property 315
IndexSearch.Current Property 316
IndexSearch.CustomOrderBy Property 316
IndexSearch.EkItems Property 316
IndexSearch.EmptyResultMsg Property 316
IndexSearch.EncodeContentHtml Property 316
IndexSearch.FolderId Property 317
IndexSearch.FormName Property 317
IndexSearch.IsSynchronized Property 317
IndexSearch.Labelafter Property 318
IndexSearch.Labelbefore Property 318

IndexSearch.LabelBetween Property 318
IndexSearch.LabelContains Property 319
IndexSearch.LabeldateBetween Property 319
IndexSearch.LabelEqual Property 319
IndexSearch.LabelExactPhrase Property 320
IndexSearch.LabelGreaterThan Property 320
IndexSearch.Labelinputdate Property 320
IndexSearch.Labelinputnumber Property 320
IndexSearch.LabelLessThan Property 321
IndexSearch.Labelon Property 321
IndexSearch.LabelSelect Property 321
IndexSearch.LinkTarget Property 322
IndexSearch.MaxResults Property 322
IndexSearch.OrderBy Property 322
IndexSearch.OrderByDirection Property 323
IndexSearch.Recursive Property 323
IndexSearch.SearchParmXML Property 323
IndexSearch.ShowSearchBoxAlways Property 324
IndexSearch.SyncRoot Property 324
IndexSearch.Weighted Property 324
IndexSearch.XmlConfigId Property 324
IndexSearch.XmlDoc Property 325
Invite Class 325
Invite Methods 325
Invite.Fill Method 325
Invite.ToString Method 326
Invite.Wrap Method 326
Invite Properties 326
Invite.DisplayXslt Property 326
Invite.FriendMessageId Property 327
Invite.InviteMessageId Property 327
Invite.OptionalMessageText Property 327
Invite.Stylesheet Property 328
Invite.XmlDoc Property 328
LanguageAPI Class 328
LanguageAPI.LanguageAPI Constructor 328
LanguageAPI Methods 328
LanguageAPI.Fill Method 329
LanguageAPI.GetLanguage Method 329

LanguageAPI.GetLanguageID Method 329
IsValid Method 329
LanguageAPI Properties 330
LanguageAPI.CurrentLanguage Property 330
LanguageAPI.CurrentLanguageID Property 330
LanguageAPI.DefaultLanguage Property 331
LanguageAPI.DefaultLanguageID Property 331
LanguageAPI.LangaugeIdList Property 331
LanguageAPI.LangaugeTitleList Property 331
LanguageAPI.MultiLanguageEnabled Property 331
LanguageAPI.OverrideDefault Property 332
LanguageAPI.SiteLanguage Property 332
LanguageAPI.SiteLanguageID Property 332
LanguageSelect Class 333
LanguageSelect Methods 333
LanguageSelect.Fill Method 333
LanguageSelect.Wrap Method 333
LanguageSelect Properties 334
LanguageSelect.LabelName Property 334
LanguageSelect.Text Property 334
LanguageSelect.WrapTag Property 334
ListSummary Class 335
ListSummary Methods 335
ListSummary.CopyTo Method 335
ListSummary.Fill Method 336
ListSummary.GetEnumerator Method 336
ListSummary.GetRssFeed Method 336
ListSummary.GetView Method 336
ListSummary.GetViewNames Method 337
ListSummary.MoveNext Method 337
ListSummary.Reset Method 337
ListSummary.this Indexer 337
ListSummary.ToString Method 338
ListSummary.Wrap Method 338
ListSummary Properties 338
ListSummary.AddText Property 339
ListSummary.CacheInterval Property 340
ListSummary.ContentParameter Property 340
ListSummary.ContentType Property 340

ListSummary.Count Property 341
ListSummary.Current Property 341
ListSummary.Direction Property 341
ListSummary.EkItems Property 342
ListSummary.EnablePaging Property 342
ListSummary.FolderDescription Property 342
ListSummary.FolderID Property 343
ListSummary.FolderName Property 343
ListSummary.FolderPath Property 343
ListSummary.GetHtml Property 343
ListSummary.IncludeIcons Property 344
ListSummary.IsSynchronized Property 344
ListSummary.LinkTarget Property 344
ListSummary.MarkupLanguage Property 345
ListSummary.MaxResults Property 345
ListSummary.OrderByDirection Property 345
ListSummary.OrderKey Property 346
ListSummary.Random Property 346
ListSummary.Recursive Property 346
ListSummary.SelTaxonomyID Property 347
ListSummary.SuppressWrapperTags Property 347
ListSummary.SyncRoot Property 347
ListSummary.XmlDoc Property 347
Login Class 347
Login Methods 348
Login.Fill Method 348
Login Properties 348
Login.AutoAddType Property 348
Login.AutoLogin Property 349
Login.OnlyAllowMemberLogin Property 349
Login.Password Property 349
Login.PromptLogout Property 349
Login.SuppressHelpButton Property 350
Login.Text Property 350
Login.UserName Property 350
Map Class 350
Map Methods 351
Map.Fill Method 351
Map.GetCallbackResult Method 351

Map.RaiseCallbackEvent Method 351
Map Properties 351
Map.Address Property 352
Map.ContentId Property 353
Map.CustomIcon Property 353
Map.DateSearch Property 353
Map.DisplayTextResult Property 353
Map.DistanceUnit Property 354
Map.EnableSearchResult Property 354
Map.FolderId Property 354
Map.ForceCategory Property 355
Map.GeoControl Property 355
Map.Latitude Property 355
Map.Longitude Property 356
Map.MapCategory Property 356
Map.MapProvider Property 356
Map.MapStyle Property 356
Map.MarkupLanguage Property 357
Map.MinZoomLevel Property 357
Map.PageSize Property 357
Map.Recursive Property 357
Map.SearchData Property 358
Map.SelectedCategoryLogicalType Property 358
Map.StartZoomLevel Property 358
Map.Stylesheet Property 358
Map.TemplateParamName Property 359
Map.TemplateUserProfile Property 359
Map.TypeControl Property 359
Map.ZoomControl Property 359
Membership Class 360
Membership Methods 360
Membership.Fill Method 360
Membership Properties 360
Membership.AllowAvatarUpload Property 361
Membership.AutoLogin Property 362
Membership.BoardID Property 362
Membership.DisplayMode Property 362
Membership.DynamicParameter Property 363
Membership.JavascriptEditorHeight Property 363

Membership.JavascriptEditorWidth Property 363
Membership.RedirectFailedURL Property 364
Membership.RedirectSuccessURL Property 364
Membership.RegisterButtonImg Property 364
Membership.RegisterButtonText Property 364
Membership.ResetButtonImg Property 365
Membership.ResetButtonText Property 365
Membership.ResetPasswordMessageId Property 365
Membership.SaveButtonImg Property 365
Membership.SaveButtonText Property 366
Membership.ShowExtended Property 366
Membership.ShowTerms Property 366
Membership.TaxonomyId Property 366
Membership.Text Property 367
Membership.UserExistsMessage Property 367
Membership.UserID Property 367
Membership.UserSuccessMessage Property 367
Membership.UserUpdateSuccessMessage Property 368
Menu Class 368
Menu Methods 368
Menu.Fill Method 368
Menu.Wrap Method 369
Menu Properties 369
Menu.CacheInterval Property 369
Menu.DefaultMenuID Property 370
Menu.DisplayXslt Property 370
Menu.DynamicParameter Property 370
Menu.Stylesheet Property 371
Menu.SuppressAddEdit Property 371
MessageBoard Class 371
MessageBoard Methods 372
MessageBoard.CustomSpamMethod Method 372
MessageBoard.Fill Method 372
MessageBoard.RaiseCallbackEvent Method 372
MessageBoard.ToString Method 372
MessageBoard.Wrap Method 373
MessageBoard Properties 373
MessageBoard.CacheInterval Property 374
MessageBoard.CommentBoxLocation Property 374

MessageBoard.DefaultObjectID Property 374
MessageBoard.DisplayXslt Property 375
MessageBoard.DynamicObjectParameter Property 375
MessageBoard.EnablePaging Property 375
MessageBoard.MarkupLanguage Property 376
MessageBoard.MaxResults Property 376
MessageBoard.Moderate Property 377
MessageBoard.ObjectType Property 377
MessageBoard.ProfileParamName Property 377
MessageBoard.ShowMaxCharacter Property 378
MessageBoard.SpamControlType Property 378
MessageBoard.SpamTimeSpan Property 378
MessageBoard.Stylesheet Property 379
MessageBoard.TemplateUserProfile Property 379
MessageBoard.XmlDoc Property 379
Messaging Class 380
Messaging Methods 380
Messaging.Fill Method 380
Messaging.ToString Method 380
Messaging.Wrap Method 380
Messaging Properties 381
Messaging.DisplayXslt Property 381
Messaging.EnablePreSearch Property 382
Messaging.FriendsOnly Property 382
Messaging.GroupParamName Property 382
Messaging.JavascriptEditorHeight Property 383
Messaging.JavascriptEditorHTMLMode Property 383
Messaging.JavascriptEditorToolbar Property 383
Messaging.JavascriptEditorWidth Property 384
Messaging.ProfileParamName Property 384
Messaging.ProfileUrl Property 385
Messaging.RecipientParamName Property 385
Messaging.ShowSearchOptions Property 385
Messaging.Stylesheet Property 386
Messaging.UserSearchDisplayXslt Property 386
Messaging.XmlDoc Property 386
MetaData Class 386
MetaData Methods 387
MetaData.Fill Method 387

MetaData Properties 387
MetaData.CacheInterval Property 388
MetaData.DefaultContentID Property 388
MetaData.DefaultItemList Property 389
MetaData.DynamicParameter Property 389
MetaData.GenerateDublinCore Property 389
MetaDataList Class 390
MetaDataList Methods 390
MetaDataList.CopyTo Method 391
MetaDataList.Fill Method 391
MetaDataList.GetEnumerator Method 391
MetaDataList.MoveNext Method 391
MetaDataList.Reset Method 391
MetaDataList.this Indexer 391
MetaDataList.Wrap Method 392
MetaDataList Properties 392
MetaDataList.CacheInterval Property 393
MetaDataList.ContentType Property 394
MetaDataList.Count Property 394
MetaDataList.Current Property 395
MetaDataList.EkItems Property 395
MetaDataList.EnablePaging Property 395
MetaDataList.ExactPhrase Property 395
MetaDataList.FolderID Property 396
MetaDataList.GetHtml Property 396
MetaDataList.IncludeIcons Property 397
MetaDataList.IsSynchronized Property 397
MetaDataList.KeyWordName Property 397
MetaDataList.KeyWordValue Property 398
MetaDataList.KeyWordValueMatchAll Property 398
MetaDataList.KeyWordValueSeparator Property 398
MetaDataList.LinkTarget Property 399
MetaDataList.MarkupLanguage Property 399
MetaDataList.MaxNumber Property 399
MetaDataList.OrderBy Property 400
MetaDataList.Recursive Property 400
MetaDataList.SortOrder Property 401
MetaDataList.SuppressWrapperTags Property 401
MetaDataList.SyncRoot Property 401

MetaDataList.XmlDoc Property 401
MicroMessaging Class 401
MicroMessaging Methods 402
MicroMessaging.CustomSpamMethod Method 402
MicroMessaging.Fill Method 402
MicroMessaging.RaiseCallbackEvent Method 402
MicroMessaging.ToString Method 403
MicroMessaging.Wrap Method 403
MicroMessaging Properties 403
MicroMessaging.CacheInterval Property 404
MicroMessaging.DefaultObjectID Property 404
MicroMessaging.DisplayXslt Property 405
MicroMessaging.DynamicObjectParameter Property 405
MicroMessaging.EnablePaging Property 405
MicroMessaging.MaxResults Property 406
MicroMessaging.MaxSearchResults Property 406
MicroMessaging.Mode Property 407
MicroMessaging.ProfileParamName Property 407
MicroMessaging.SpamControlType Property 407
MicroMessaging.SpamTimeSpan Property 408
MicroMessaging.Stylesheet Property 408
MicroMessaging.TemplateMessage Property 408
MicroMessaging.TemplateUserProfile Property 409
MicroMessaging.XmlDoc Property 409
MyAccount Class 409
MyAccount Methods 410
MyAccount.Fill Method 410
MyAccount.ToString Method 410
MyAccount.Wrap Method 410
MyAccount Properties 410
MyAccount.CustomPropertyID Property 411
MyAccount.DefaultCountryId Property 411
MyAccount.DisplayXslt Property 412
MyAccount.EkItem Property 412
MyAccount.LoadingImage Property 412
MyAccount.RedirectUrl Property 413
MyAccount.Stylesheet Property 413
MyAccount.UseMsAjax Property 413
OrderList Class 414

OrderList Methods 414
OrderList.Fill Method 414
OrderList.ToString Method 415
OrderList.Wrap Method 415
OrderList Properties 415
OrderList.CacheInterval Property 415
OrderList.DefaultUserID Property 416
OrderList.DisplayXslt Property 416
OrderList.DynamicOrderParameter Property 417
OrderList.DynamicProductParameter Property 417
OrderList.EkItem Property 417
OrderList.PageSize Property 418
OrderList.Stylesheet Property 418
OrderList.TemplateProduct Property 418
PersonalizationManager Class 419
PersonalizationManager Methods 419
PersonalizationManager.Fill Method 419
PersonalizationManager Properties 420
PersonalizationManager.AddContentExitText Property 420
PersonalizationManager.AddContentText Property 420
PersonalizationManager.CurrentUserButtonText Property 421
PersonalizationManager.Display Property 421
PersonalizationManager.HeaderBackColor Property 421
PersonalizationManager.HeaderText Property 422
PersonalizationManager.PersonalizeExitText Property 422
PersonalizationManager.PersonalizeScopeText Property 422
PersonalizationManager.PersonalizeStartText Property 423
PersonalizationManager.PublicUserButtonText Property 423
PersonalizationManager.ResetLinkText Property 423
PersonalizationManager.ResetLinkToolTip Property 424
PersonalizationManager.UrlCompareMode Property 424
PhotoGallery Class 425
PhotoGallery Methods 425
PhotoGallery.Fill Method 425
Poll Class 425
Poll Methods 426
Poll.Fill Method 426
Poll.WrapPoll Method 426
Poll Properties 427

Poll.EnableAjax Property 427
Poll.PollID Property 427
PostHistory Class 428
PostHistory Methods 428
PostHistory.Fill Method 428
PostHistory.ToString Method 428
PostHistory.Wrap Method 429
PostHistory Properties 429
PostHistory.BoardID Property 429
PostHistory.CacheInterval Property 429
PostHistory.MaxNumber Property 430
PostHistory.URLPath Property 430
PostHistory.UserID Property 430
PostLoginEventArgs Class 431
Product Class 431
Product Methods 431
Product.Fill Method 432
Product.RedirectToBasketPage Method 432
Product.ToString Method 432
Product.Wrap Method 432
Product Properties 432
Product.CacheInterval Property 433
Product.DefaultProductID Property 434
Product.DisplayXslt Property 434
Product.DynamicParameter Property 434
Product.EkItems Property 435
Product.ImageGallery Property 435
Product.OverrideXslt Property 436
Product.ShowAddToCart Property 436
Product.Stylesheet Property 436
Product.TemplateCart Property 437
ProductList Class 437
ProductList Methods 437
ProductList.Fill Method 438
ProductList.ToString Method 438
ProductList.Wrap Method 438
ProductList Properties 438
ProductList.DisplayXslt Property 439
ProductList.DynamicParameter Property 440

ProductList.EkItems Property 440
ProductList.IdList Property 440
ProductList.LoadingImage Property 441
ProductList.PageSize Property 441
ProductList.PageSpan Property 441
ProductList.ProductCriteria Property 442
ProductList.Recursive Property 442
ProductList.SortMode Property 442
ProductList.SourceId Property 443
ProductList.SourceType Property 443
ProductList.Stylesheet Property 444
ProductList.TaxonomyDepth Property 444
ProductList.TemplateProduct Property 445
ProductList.UseMsAjax Property 445
ProductSearch Class 445
ProductSearch.ProductSearch Constructor 446
ProductSearch Methods 446
ProductSearch.Fill Method 446
ProductSearch.GetCallbackResult Method 446
ProductSearch.RaiseCallbackEvent Method 447
ProductSearch.Wrap Method 447
ProductSearch Properties 447
ProductSearch.ButtonImgSrc Property 449
ProductSearch.ButtonText Property 450
ProductSearch.CacheInterval Property 450
ProductSearch.CatalogId Property 450
ProductSearch.CustomOrderBy Property 451
ProductSearch.DisplayXslt Property 451
ProductSearch.DynamicProductParameter Property 451
ProductSearch.EnableAdvancedLink Property 452
ProductSearch.EndCount Property 452
ProductSearch.IsInWorkArea Property 452
ProductSearch.LinkTarget Property 453
ProductSearch.LoadingImage Property 453
ProductSearch.MaxCharacters Property 453
ProductSearch.MaxTeaserLength Property 454
ProductSearch.OrderBy Property 454
ProductSearch.OrderDirection Property 455
ProductSearch.Recursive Property 455

ProductSearch.ResultCount Property 455
ProductSearch.ResultDataTable Property 456
ProductSearch.ResultsPageSize Property 456
ProductSearch.ResultTagId Property 456
ProductSearch.ResultXml Property 457
ProductSearch.SearchText Property 457
ProductSearch.ShowCategories Property 457
ProductSearch.ShowCustomSummary Property 458
ProductSearch.ShowSearchBoxAlways Property 458
ProductSearch.StartCount Property 459
ProductSearch.Stylesheet Property 459
ProductSearch.TaxCategories Property 459
ProductSearch.TaxonomyOperator Property 459
ProductSearch.TemplateCart Property 460
ProductSearch.TemplateProduct Property 460
ProductSearch.TextBoxSize Property 461
ProductSearch.TimeTaken Property 461
ProductSearch.XmlDoc Property 461
Recommendation Class 462
Recommendation Methods 462
Recommendation.Fill Method 462
Recommendation.ToString Method 462
Recommendation.Wrap Method 462
Recommendation Properties 463
Recommendation.CacheInterval Property 463
Recommendation.DefaultProductID Property 464
Recommendation.DisplayXslt Property 464
Recommendation.DynamicCartParameter Property 465
Recommendation.DynamicProductParameter Property 465
Recommendation.EkItem Property 466
Recommendation.PageSize Property 466
Recommendation.RecommendationType Property 466
Recommendation.Stylesheet Property 467
Recommendation.TemplateCart Property 467
Recommendation.TemplateProduct Property 467
RssAggregator Class 468
RssAggregator Methods 468
RssAggregator.CopyTo Method 469
RssAggregator.Fill Method 469

RssAggregator.GetEnumerator Method 469
RssAggregator.GetView Method 469
RssAggregator.GetViewNames Method 469
RssAggregator.MoveNext Method 470
RssAggregator.Reset Method 470
RssAggregator.this Indexer 470
RssAggregator Properties 470
RssAggregator.Count Property 471
RssAggregator.Current Property 471
RssAggregator.EkItems Property 471
RssAggregator.IsSynchronized Property 471
RssAggregator.LinkTarget Property 472
RssAggregator.MaxResults Property 472
RssAggregator.SyncRoot Property 472
RssAggregator.URL Property 473
RssAggregator.WrapTag Property 473
RssItemBase Class 473
RssItemBase Fields 473
RssItemBase.DatePosted Field 474
RssItemBase.Description Field 474
RssItemBase.Link Field 474
RssItemBase.Title Field 474
Search Class 474
Search.Search Constructor 475
Search Methods 475
Search.CopyTo Method 476
Search.Fill Method 476
Search.GetEnumerator Method 476
Search.LoadSearch Method 476
Search.MoveNext Method 476
Search.Reset Method 477
Search.SearchDisplayUI Method 477
Search.this Indexer 477
Search.Wrap Method 477
Search Properties 477
Search.ButtonImgSrc Property 480
Search.ButtonText Property 480
Search.ContentFieldsetLegend Property 480
Search.Count Property 481

Search.Current Property 481
Search.Display Property 481
Search.EkItems Property 481
Search.EmptyResultMsg Property 482
Search.EnableAdvancedLink Property 482
Search.EnableAjax Property 482
Search.EnableArchived Property 483
Search.EnableArchivedBtn Property 483
Search.EnableContent Property 483
Search.EnableContentBtn Property 483
Search.EnableContentFieldset Property 484
Search.EnableDMSAssets Property 484
Search.EnableDMSAssetsBtn Property 484
Search.EnableFolderFieldset Property 485
Search.EnableForms Property 485
Search.EnableFormsBtn Property 485
Search.EnableLibFiles Property 486
Search.EnableLibFilesBtn Property 486
Search.EnableLibHyperlinks Property 486
Search.EnableLibHyperlinksBtn Property 486
Search.EnableLibImages Property 487
Search.EnableLibImagesBtn Property 487
Search.EnableLibraryFieldset Property 487
Search.FolderFieldsetLegend Property 488
Search.FolderID Property 488
Search.FolderPath Property 488
Search.IncludeIcons Property 489
Search.IsSynchronized Property 489
Search.LibraryFieldsetLegend Property 489
Search.LinkTarget Property 490
Search.MaxCharacters Property 490
Search.MaxResults Property 490
Search.MaxTeaserLength Property 491
Search.OrderBy Property 491
Search.OrderDirection Property 491
Search.PostResultsAt Property 492
Search.Recursive Property 492
Search.RemoveTeaserHtml Property 492
Search.ResultTagId Property 493

Search.SearchText Property 493
Search.SearchType Property 493
Search.ShowDate Property 493
Search.ShowExtendedSearch Property 494
Search.ShowSearchBoxAlways Property 494
Search.ShowSearchOptions Property 494
Search.Stylesheet Property 495
Search.SyncRoot Property 495
Search.TargetFolderDescriptionList Property 495
Search.TextBoxSize Property 496
Search.XmlDoc Property 496
Sitemap Class 496
Sitemap Methods 496
Sitemap.Fill Method 496
Sitemap Properties 497
Sitemap.CacheInterval Property 497
Sitemap.ClassName Property 498
Sitemap.DisplayType Property 498
Sitemap.DisplayXslt Property 498
Sitemap.FlatTopLevel Property 499
Sitemap.FolderID Property 499
Sitemap.MaxLevel Property 499
Sitemap.StartingLevel Property 500
Sitemap.Stylesheet Property 500
Sitemap.XmlDoc Property 500
SmartMenu Class 500
SmartMenu Properties 501
SmartMenu.AutoCollapseBranches Property 501
SmartMenu.DisplayXslt Property 502
SmartMenu.Enable508Compliance Property 502
SmartMenu.Enable508NavigationHeadings Property 502
SmartMenu.EnableMouseOverPopUp Property 503
SmartMenu.EnableSmartOpen Property 503
SmartMenu.EnableSubmenuLinkPathMatching Property 504
SmartMenu.EnableSubmenuTemplateMatching Property 504
SmartMenu.LaunchLinksInNewWindow Property 504
SmartMenu.LogInfo Property 505
SmartMenu.ShowRootFolder Property 505
SmartMenu.SingleEditButton Property 505

SmartMenu.StartCollapsed Property 506
SmartMenu.StartWithRootCollapsed Property 506
SmartMenu.Stylesheet Property 506
SocialBar Class 507
SocialBar Methods 507
SocialBar.Fill Method 507
SocialBar.GetCallbackResult Method 508
SocialBar.ToString Method 508
SocialBar.Wrap Method 508
SocialBar Properties 508
SocialBar.CacheInterval Property 509
SocialBar.DefaultObjectID Property 509
SocialBar.DisplayXslt Property 510
SocialBar.DynamicObjectParameter Property 510
SocialBar.EkItems Property 510
SocialBar.IncludeIcons Property 511
SocialBar.InviteURL Property 511
SocialBar.Items Property 511
SocialBar.MarkupLanguage Property 512
SocialBar.MessagingURL Property 512
SocialBar.ObjectType Property 512
SocialBar.Stylesheet Property 512
SocialBar.SuppressWrapperTags Property 513
SocialBar.XmlDoc Property 513
TagCloud Class 513
TagCloud Methods 514
TagCloud.Fill Method 514
TagCloud.ToString Method 514
TagCloud.Wrap Method 514
TagCloud Properties 514
TagCloud.CacheInterval Property 515
TagCloud.DisplayXslt Property 516
TagCloud.DynamicParameter Property 516
TagCloud.MarkupLanguage Property 516
TagCloud.MaxTagsDisplayed Property 517
TagCloud.OrderBy Property 517
TagCloud.OrderByDirection Property 517
TagCloud.Stylesheet Property 518
TagCloud.TagTemplate Property 518

TagCloud.TagTemplateTarget Property 519
TagCloud.TagType Property 519
TagCloud.XmlDoc Property 519
UserProfile Class 520
UserProfile Methods 520
UserProfile.Fill Method 520
UserProfile.ToString Method 520
UserProfile.Wrap Method 520
UserProfile Properties 521
UserProfile.AvatarHeight Property 521
UserProfile.AvatarWidth Property 521
UserProfile.CacheInterval Property 522
UserProfile.DefaultUserID Property 522
UserProfile.DisplayXslt Property 523
UserProfile.DynamicParameter Property 523
UserProfile.EnableEmailDisplay Property 523
UserProfile.Stylesheet Property 524
UserProfile.TagTemplate Property 524
UserProfile.TagTemplateTarget Property 524
UserProfile.TaxonomyId Property 524
UserProfile.XmlDoc Property 525
WebCalendar Class 525
WebCalendar Methods 525
WebCalendar.Fill Method 526
WebCalendar.Wrap Method 526
WebCalendar Properties 526
WebCalendar.AllowEventEditing Property 526
WebCalendar.CssFile Property 527
WebCalendar.DataSource Property 527
WebCalendar.DefaultDisplayType Property 527
WebCalendar.DisplayTemplatePath Property 527
WebCalendar.DisplayType Property 528
WebCalendar.DynamicParameter Property 528
WebCalendar.RequestInformationRef Property 528
WebCalendar.Text Property 529
WebSearch Class 529
WebSearch.WebSearch Constructor 529
WebSearch Methods 529
WebSearch.Fill Method 530

WebSearch.GetCallbackResult Method 530
WebSearch.LoadSearch Method 530
WebSearch.RaiseCallbackEvent Method 530
WebSearch.Wrap Method 531
WebSearch Properties 531
WebSearch.ButtonImgSrc Property 534
WebSearch.ButtonText Property 534
WebSearch.CustomOrderBy Property 534
WebSearch.CustomSearch Property 535
WebSearch.DisableForumSearch Property 535
WebSearch.DisplayXslt Property 535
WebSearch.DynamicContentTemplate Property 536
WebSearch.EnableAdvancedLink Property 536
WebSearch.EndCount Property 536
WebSearch.FolderID Property 536
WebSearch.IsInWorkArea Property 537
WebSearch.LinkTarget Property 537
WebSearch.MarkupLanguage Property 538
WebSearch.MaxCharacters Property 538
WebSearch.MaxTeaserLength Property 538
WebSearch.MultipleFolders Property 539
WebSearch.OrderBy Property 539
WebSearch.OrderDirection Property 539
WebSearch.Recursive Property 540
WebSearch.RemoveTeaserHtml Property 540
WebSearch.ResultCount Property 540
WebSearch.ResultDataTable Property 541
WebSearch.ResultsPageSize Property 541
WebSearch.ResultTagId Property 542
WebSearch.ResultXml Property 542
WebSearch.SearchFor Property 542
WebSearch.SearchSynonyms Property 543
WebSearch.SearchText Property 543
WebSearch.ShowCategories Property 544
WebSearch.ShowCustomSummary Property 544
WebSearch.ShowSearchBoxAlways Property 544
WebSearch.ShowSearchOptions Property 545
WebSearch.ShowSuggestedResults Property 545
WebSearch.StartCount Property 546

WebSearch.Stylesheet Property 546
WebSearch.TaxCategories Property 546
WebSearch.TaxonomyOperator Property 546
WebSearch.TextBoxSize Property 547
WebSearch.TimeTaken Property 547
WebSearch.XmlDoc Property 547
Structs, Records, Enums 548
Ektron.Cms.Controls.CommunityDocumentsObjectType Enumeration 548
Ektron.Cms.Controls.WebCalendarForms Namespace 548
Classes 549
AdvancedEditTemplate Class 549
AdvancedFormBase Class 549
AdvancedInsertTemplate Class 550
AdvancedTemplate Class 550
AppointmentTemplate Class 550
DataKeys Class 551
DisplayTemplate Class 551
SchedulerStyles Class 551

Plug-in Extension API 552
Ektron.Cms.Extensibility Namespace 552
Classes 553
ActionRecipientInfo Class 553
ActionRecipientInfo.ActionRecipientInfo Constructor 554
ActionRecipientInfo Properties 554
ActionRecipientInfo.DisplayName Property 554
ActionRecipientInfo.Email Property 554
ActionRecipientInfo.FirstName Property 555
ActionRecipientInfo.IsUser Property 555
ActionRecipientInfo.LastName Property 555
ActionRecipientInfo.UserId Property 555
ActionRequestInfo Class 555
ActionRequestInfo.ActionRequestInfo Constructor 556
ActionRequestInfo Enumerations 556
Ektron.Cms.Extensibility.ActionRequestInfo.ActionRequestType Enumeration 556
ActionRequestInfo Properties 556
ActionRequestInfo.ErrorMessage Property 557
ActionRequestInfo.Id Property 557
ActionRequestInfo.Recipients Property 557
ActionRequestInfo.SenderId Property 557

ActionRequestInfo.Type Property 557
ContentInfo Class 558
ContentInfo Properties 558
ContentInfo.AssetID Property 558
ContentInfo.Comments Property 559
ContentInfo.ErrorMessage Property 559
ContentInfo.FileAndPath Property 559
ContentInfo.FileExtension Property 559
ContentInfo.FileName Property 559
ContentInfo.FolderID Property 559
ContentInfo.Html Property 560
ContentInfo.ID Property 560
ContentInfo.Language Property 560
ContentInfo.MimeType Property 560
ContentInfo.Status Property 560
ContentInfo.Title Property 560
ContentInfo.Type Property 561
ContentInfo.UserID Property 561
ExtensionAuthor Class 561
ExtensionAuthor.ExtensionAuthor Constructor 561
ExtensionAuthor Methods 561
ExtensionAuthor.ToString Method 562
ExtensionAuthor Properties 562
ExtensionAuthor.Author Property 562
ExtensionDescription Class 562
ExtensionDescription.ExtensionDescription Constructor 563
ExtensionDescription Methods 563
ExtensionDescription.ToString Method 563
ExtensionDescription Properties 563
ExtensionDescription.Desc Property 563
ExtensionEvent Class 563
ExtensionEvent Methods 564
ExtensionEvent.GetActiveDirectoryUserProxy Method 566
ExtensionEvent.GetAssetProxy Method 566
ExtensionEvent.GetBlogProxy Method 566
ExtensionEvent.GetBusinessRulesProxy Method 567
ExtensionEvent.GetCalendarEventTypeProxy Method 567
ExtensionEvent.GetCalendarProxy Method 567
ExtensionEvent.GetCommunityGroupProxy Method 567

ExtensionEvent.GetContentProxy Method 567
ExtensionEvent.GetContentRatingProxy Method 568
ExtensionEvent.GetCustomFieldsProxy Method 568
ExtensionEvent.GetFavoritesProxy Method 568
ExtensionEvent.GetFlagProxy Method 568
ExtensionEvent.GetFolderProxy Method 569
ExtensionEvent.GetFontProxy Method 569
ExtensionEvent.GetFormProxy Method 569
ExtensionEvent.GetFriendsProxy Method 569
ExtensionEvent.GetLibraryProxy Method 570
ExtensionEvent.GetMessageBoardProxy Method 570
ExtensionEvent.GetMetadataProxy Method 570
ExtensionEvent.GetObjectStateAsXml Method 570
ExtensionEvent.GetPermissionsProxy Method 570
ExtensionEvent.GetSearchManagerProxy Method 571
ExtensionEvent.GetSiteMapProxy Method 571
ExtensionEvent.GetSiteProxy Method 571
ExtensionEvent.GetTagsProxy Method 571
ExtensionEvent.GetTaskCategoryProxy Method 572
ExtensionEvent.GetTaskCategoryTypeProxy Method 572
ExtensionEvent.GetTaskProxy Method 572
ExtensionEvent.GetTaxonomyProxy Method 572
ExtensionEvent.GetThreadedDiscussionProxy Method 573
ExtensionEvent.GetUserProxy Method 573
ExtensionEvent.OnAfterActionRequestSent Method 573
ExtensionEvent.OnAfterContentAdd Method 573
ExtensionEvent.OnAfterContentDelete Method 574
ExtensionEvent.OnAfterContentEdit Method 574
ExtensionEvent.OnAfterContentFlagged Method 574
ExtensionEvent.OnAfterFolderAdd Method 574
ExtensionEvent.OnAfterFolderDelete Method 574
ExtensionEvent.OnAfterFolderEdit Method 575
ExtensionEvent.OnAfterFormSubmit Method 575
ExtensionEvent.OnAfterPublish Method 575
ExtensionEvent.OnAfterReviewRejected Method 575
ExtensionEvent.OnAfterUserAdd Method 576
ExtensionEvent.OnAfterUserDelete Method 576
ExtensionEvent.OnAfterUserEdit Method 576
ExtensionEvent.OnAfterUserLogin Method 576

ExtensionEvent.OnBeforeActionRequestSent Method 577
ExtensionEvent.OnBeforeContentAdd Method 577
ExtensionEvent.OnBeforeContentDelete Method 577
ExtensionEvent.OnBeforeContentEdit Method 577
ExtensionEvent.OnBeforeContentFlagged Method 578
ExtensionEvent.OnBeforeFolderAdd Method 578
ExtensionEvent.OnBeforeFolderDelete Method 578
ExtensionEvent.OnBeforeFolderEdit Method 578
ExtensionEvent.OnBeforeFormSubmit Method 578
ExtensionEvent.OnBeforePublish Method 579
ExtensionEvent.OnBeforeReviewRejected Method 579
ExtensionEvent.OnBeforeUserAdd Method 579
ExtensionEvent.OnBeforeUserDelete Method 579
ExtensionEvent.OnBeforeUserEdit Method 580
ExtensionEvent.OnBeforeUserLogin Method 580
ExtensionEvent Properties 580
ExtensionEvent.Content Property 580
ExtensionEvent.Folder Property 581
ExtensionEvent.Form Property 581
ExtensionEvent.Site Property 581
ExtensionEvent.User Property 581
FlagInfo Class 581
FlagInfo Properties 582
FlagInfo.ContentHtml Property 582
FlagInfo.ContentID Property 582
FlagInfo.ContentLanguage Property 583
FlagInfo.ContentTitle Property 583
FlagInfo.EntryId Property 583
FlagInfo.ErrorMessage Property 583
FlagInfo.FlagComment Property 583
FlagInfo.FlagDate Property 583
FlagInfo.FlagId Property 584
FlagInfo.FlagName Property 584
FlagInfo.UserAvatar Property 584
FlagInfo.UserDisplayName Property 584
FlagInfo.UserFirstName Property 584
FlagInfo.UserID Property 584
FlagInfo.UserLastName Property 585
FlagInfo.UserName Property 585

FlagInfo.VisitorID Property 585
FolderInfo Class 585
FolderInfo Properties 585
FolderInfo.Description Property 586
FolderInfo.ErrorMessage Property 586
FolderInfo.ID Property 586
FolderInfo.Name Property 586
FolderInfo.ParentID Property 586
FolderInfo.ReplicationState Property 586
FolderInfo.StyleSheet Property 587
FolderInfo.Type Property 587
FormField Class 587
FormField.FormField Constructor 587
FormField Properties 588
FormField.FieldName Property 588
FormField.FieldValue Property 588
FormInfo Class 588
FormInfo Methods 588
FormInfo.this Indexer 588
FormInfo Properties 589
FormInfo.ErrorMessage Property 589
FormInfo.FormDescription Property 589
FormInfo.FormFields Property 589
FormInfo.FormTitle Property 589
MessageBoardInfo Class 590
MessageBoardInfo.MessageBoardInfo Constructor 590
MessageBoardInfo Properties 590
MessageBoardInfo.ErrorMessage Property 590
RequestInfo Class 591
ReviewInfo Class 591
ReviewInfo Properties 591
ReviewInfo.AverageRating Property 592
ReviewInfo.ContentHtml Property 592
ReviewInfo.ContentID Property 592
ReviewInfo.ContentLanguage Property 592
ReviewInfo.ContentTitle Property 592
ReviewInfo.ErrorMessage Property 593
ReviewInfo.ID Property 593
ReviewInfo.NumberofRatings Property 593

ReviewInfo.Rating Property 593
ReviewInfo.RatingDate Property 593
ReviewInfo.State Property 593
ReviewInfo.UserAvatar Property 594
ReviewInfo.UserComments Property 594
ReviewInfo.UserDisplayName Property 594
ReviewInfo.UserFirstName Property 594
ReviewInfo.UserID Property 594
ReviewInfo.UserLastName Property 594
ReviewInfo.UserName Property 595
ReviewInfo.VisitorID Property 595
SiteInfo Class 595
SiteInfo Properties 595
SiteInfo.AppPath Property 595
SiteInfo.IsSSL Property 596
SiteInfo.Name Property 596
SiteInfo.PhysicalSitePath Property 596
SiteInfo.Port Property 596
SiteInfo.Url Property 596
UserInfo Class 596
UserInfo Properties 597
UserInfo.Addres Property 597
UserInfo.DisplayName Property 597
UserInfo.Editor Property 598
UserInfo.EmailAddress Property 598
UserInfo.ErrorMessage Property 598
UserInfo.FirstName Property 598
UserInfo.ID Property 598
UserInfo.Language Property 598
UserInfo.LastName Property 599
UserInfo.Latitude Property 599
UserInfo.Longitude Property 599
UserInfo.Notifications Property 599
UserInfo.Type Property 599
UserInfo.UserName Property 599

.NET Assemblies 600
eCommerce 601
Classes 601
AddressApi Class 602

AddressApi Methods 602
Add Method 603
AddressApi.Add Method (AddressData) 603
AddressApi.Add Method (AddressData, long) 603
AddressApi.Delete Method 604
AddressApi.GetItem Method 604
AddressApi.GetList Method 604
AddressApi.Update Method 604
AddressApi.UpdateOrderAddress Method 605
BasketApi Class 605
BasketApi Methods 605
CalculateBasket Method 606
BasketApi.CalculateBasket Method (Basket) 606
BasketApi.CalculateBasket Method (Basket, AddressData) 606
BasketApi.CalculateBasket Method (long) 607
BasketApi.CalculateBasket Method (long, AddressData) 607
BasketApi.CreateNewBasket Method 607
BasketApi.Delete Method 608
BasketApi.Empty Method 608
GetDefaultBasket Method 608
BasketApi.GetDefaultBasket Method () 609
BasketApi.GetDefaultBasket Method (long) 609
BasketApi.GetDefaultBasket Method (string) 609
BasketApi.GetItem Method 609
BasketApi.GetList Method 610
SetDefaultBasket Method 610
BasketApi.SetDefaultBasket Method (long) 610
BasketApi.SetDefaultBasket Method (long, long) 611
CatalogEntryApi Class 611
CatalogEntryApi Methods 611
CatalogEntryApi.Add Method 612
CatalogEntryApi.AddAndCheckIn Method 612
CatalogEntryApi.AddAndPublish Method 612
CatalogEntryApi.AddAndSave Method 612
CatalogEntryApi.Checkout Method 613
CatalogEntryApi.Delete Method 613
CatalogEntryApi.GetAttributeEntryList Method 613
GetItem Method 613
CatalogEntryApi.GetItem Method (long) 614

CatalogEntryApi.GetItem Method (long, long) 614
CatalogEntryApi.GetItemEdit Method 614
CatalogEntryApi.GetList Method 615
CatalogEntryApi.GetTaxonomyEntryList Method 615
CatalogEntryApi.Restore Method 615
CatalogEntryApi.UndoCheckOut Method 615
CatalogEntryApi.Update Method 616
CatalogEntryApi.UpdateAndCheckIn Method 616
CatalogEntryApi.UpdateAndPublish Method 616
ContractApi Class 616
ContractApi Methods 617
ContractApi.Add Method 617
ContractApi.Delete Method 617
ContractApi.GetItem Method 617
ContractApi.GetList Method 618
ContractApi.Update Method 618
CountryApi Class 618
CountryApi Methods 618
CountryApi.Add Method 619
CountryApi.CanDelete Method 619
CountryApi.Delete Method 619
CountryApi.GetItem Method 620
CountryApi.GetList Method 620
CountryApi.Update Method 620
CouponApi Class 620
CouponApi Methods 621
CouponApi.Add Method 621
CouponApi.AddCouponToCatalog Method 622
CouponApi.AddCouponToProduct Method 622
CouponApi.AddCouponToTaxonomy Method 622
CouponApi.ApplyCouponToBasket Method 623
CalculateBasketCoupons Method 623
CouponApi.CalculateBasketCoupons Method (Basket) 623
CouponApi.CalculateBasketCoupons Method (long) 623
CouponApi.Deactivate Method 624
CouponApi.Delete Method 624
CouponApi.DeleteCouponProducts Method 624
CouponApi.GetCatalogList Method 624
CouponApi.GetCouponsAppliedToBasket Method 625

CouponApi.GetItem Method 625
CouponApi.GetList Method 625
GetProductList Method 625
CouponApi.GetProductList Method (Criteria<CouponEntryProperty>) 626
CouponApi.GetProductList Method (long) 626
CouponApi.GetTaxonomyList Method 626
CouponApi.IsCouponAppliedToProduct Method 626
CouponApi.IsCouponUsedForBasket Method 627
CouponApi.IsCouponUsedForOrder Method 627
CouponApi.SaveCouponApplications Method 627
CouponApi.Update Method 628
ValidateCoupon Method 628
CouponApi.ValidateCoupon Method (long) 628
CouponApi.ValidateCoupon Method (long, long) 628
CurrencyApi Class 629
CurrencyApi Methods 629
CurrencyApi.Add Method 629
CurrencyApi.CanDelete Method 629
CurrencyApi.Delete Method 630
CurrencyApi.GetItem Method 630
CurrencyApi.GetList Method 630
CurrencyApi.Update Method 630
CustomerApi Class 631
CustomerApi Methods 631
Add Method 631
CustomerApi.Add Method (CustomerData, AddressData) 632
CustomerApi.Add Method (CustomerData, AddressData, AddressData) 632
CustomerApi.ChangeBillingAddress Method 632
CustomerApi.ChangeShippingAddress Method 633
CustomerApi.Delete Method 633
CustomerApi.GetItem Method 633
CustomerApi.GetList Method 633
CustomerApi.GetOrderList Method 634
CustomerApi.GetOrderTotal Method 634
CustomerApi.Update Method 634
ExchangeRateApi Class 634
ExchangeRateApi Methods 635
ExchangeRateApi.Add Method 635
ExchangeRateApi.ConvertPrice Method 635

ExchangeRateApi.Delete Method 636
ExchangeRateApi.GetCurrentExchangeRate Method 636
ExchangeRateApi.GetCurrentList Method 636
ExchangeRateApi.GetList Method 636
ExchangeRateApi.Update Method 637
InventoryApi Class 637
InventoryApi Methods 637
InventoryApi.SaveInventory Method 637
OrderApi Class 638
OrderApi Methods 638
Capture Method 639
OrderApi.Capture Method (long) 639
OrderApi.Capture Method (long, long) 639
OrderApi.GetCustomerOrderList Method 639
OrderApi.GetItem Method 640
GetList Method 640
OrderApi.GetList Method (Criteria<OrderProperty>) 640
OrderApi.GetList Method (DateTime, DateTime, PagingInfo) 641
GetOnHoldOrderList Method 641
OrderApi.GetOnHoldOrderList Method (PagingInfo) 641
OrderApi.GetOnHoldOrderList Method (PagingInfo, OrderProperty, EkEnumeration.OrderByDirection) 641
GetOrderMessage Method 642
OrderApi.GetOrderMessage Method (EkEnumeration.EmailMessageTypes, long) 642
OrderApi.GetOrderMessage Method (long, long) 642
OrderApi.GetOrderPayment Method 643
OrderApi.GetOrderStatus Method 643
GetProductOrderList Method 643
OrderApi.GetProductOrderList Method (long, Criteria<OrderProperty>) 643
OrderApi.GetProductOrderList Method (long, PagingInfo) 644
OrderApi.GetReport Method 644
OrderApi.ItemsInStock Method 644
OrderApi.MarkAsFraud Method 645
MarkPaymentAsSettled Method 645
OrderApi.MarkPaymentAsSettled Method (long) 645
OrderApi.MarkPaymentAsSettled Method (long, long) 645
OrderApi.ProcessOrderInventory Method 646
OrderApi.RollbackOrderInventory Method 646
OrderApi.SetOrderStatus Method 646

OrderApi.SetOrderToHold Method 646
OrderApi.SetTrackingNumber Method 647
PackageApi Class 647
PackageApi Methods 647
PackageApi.Add Method 647
PackageApi.Delete Method 648
PackageApi.GetItem Method 648
PackageApi.GetList Method 648
PackageApi.Update Method 649
ProductTypeApi Class 649
ProductTypeApi Methods 649
ProductTypeApi.Add Method 650
ProductTypeApi.AddProductAttribute Method 650
ProductTypeApi.Delete Method 650
DeleteProductAttribute Method 650
ProductTypeApi.DeleteProductAttribute Method (long, System.Data.IDbTransaction) 650
ProductTypeApi.DeleteProductAttribute Method (long, bool, System.Data.IDbTransaction) 651
ProductTypeApi.GetFolderProductTypeList Method 651
GetItem Method 651
ProductTypeApi.GetItem Method (long) 651
ProductTypeApi.GetItem Method (long, bool) 652
ProductTypeApi.GetProductAttribute Method 652
ProductTypeApi.GetProductAttributeList Method 652
ProductTypeApi.Update Method 652
ProductTypeApi.UpdateProductAttribute Method 652
RecommendationApi Class 653
RecommendationApi Methods 653
RecommendationApi.Add Method 653
RecommendationApi.Delete Method 653
RecommendationApi.DeleteRecommendationByEntry Method 654
RecommendationApi.GetItem Method 654
RecommendationApi.Update Method 654
RecommendationApi.UpdateCrossSell Method 654
RecommendationApi.UpdateUpSell Method 655
RegionApi Class 655
RegionApi Methods 655
RegionApi.Add Method 655
RegionApi.CanDelete Method 656
RegionApi.Delete Method 656

RegionApi.GetItem Method 656
RegionApi.GetList Method 657
RegionApi.Update Method 657
ShippingMethodApi Class 657
ShippingMethodApi Methods 657
ShippingMethodApi.Add Method 658
ShippingMethodApi.Update Method 658
TaxApi Class 658
TaxApi Methods 658
TaxApi.Add Method 659
TaxApi.Delete Method 659
TaxApi.GetApplicableTaxRate Method 659
TaxApi.GetApplicableTaxRateList Method 660
GetItem Method 660
TaxApi.GetItem Method (long) 660
TaxApi.GetItem Method (long, long) 660
TaxApi.GetItemByCountryId Method 661
GetItemByPostalCode Method 661
TaxApi.GetItemByPostalCode Method (long, string) 661
TaxApi.GetItemByPostalCode Method (long, string, int) 661
TaxApi.GetItemByPostalId Method 662
TaxApi.GetItemByRegionId Method 662
TaxApi.GetList Method 662
TaxApi.Update Method 663
TaxClassApi Class 663
TaxClassApi Methods 663
TaxClassApi.Add Method 663
TaxClassApi.Delete Method 664
TaxClassApi.GetItem Method 664
TaxClassApi.GetList Method 664
TaxClassApi.IsUsed Method 665
TaxClassApi.Update Method 665
TaxTypeApi Class 665
TaxTypeApi Methods 665
TaxTypeApi.Add Method 666
TaxTypeApi.Delete Method 666
TaxTypeApi.GetItem Method 666
TaxTypeApi.GetList Method 666
TaxTypeApi.Update Method 667

WarehouseApi Class 667
WarehouseApi Methods 667
WarehouseApi.Add Method 668
WarehouseApi.Delete Method 668
WarehouseApi.GetDefaultWarehouse Method 668
WarehouseApi.GetItem Method 668
WarehouseApi.GetList Method 668
WarehouseApi.MarkAsDefault Method 669
WarehouseApi.Update Method 669
Ektron.Cms.API Namespace 669
Classes 670
BusinessRule Class 671
BusinessRule Methods 671
GetBusinessRules Method 671
BusinessRule.GetBusinessRules Method (String, String, String, String, String, String, String, Long) 672
BusinessRule.GetBusinessRules Method (String, String, String, String, String, String, String, String) 672
CustomFields Class 673
CustomFields Methods 673
CustomFields.AddNewEditableCustomFieldAssignments Method 674
CustomFields.BoolToYesNo Method 675
CustomFields.GetEditableCustomFieldAssignments Method 675
CustomFields.GetEnhancedMetadataArea Method 676
CustomFields.GetEnhancedMetadataScript Method 676
CustomFields.GetFieldsByFolder Method 676
GetSearchProperties Method 677
CustomFields.GetSearchProperties Method (Ektron.Cms.Common.SearchAssetDispayRequest) 677
CustomFields.GetSearchProperties Method (String, String, String, String, Collection, String, String, String) 677
CustomFields.GetSearchProperties Method (System.Collections.Specialized.NameValueCollection, Ektron.Cms.Common.SearchItemCustom, Ektron.Cms.Common.SearchItemCustom) 678
CustomFields.Populate_AssetRequestObjectFromForm Method 678
CustomFields.ProcessCustomFields Method 679
CustomFields.SearchAllAssets Method 679
CustomFields.WriteFilteredMetadataForEdit Method 679
CustomFields.WriteFilteredMetadataForView Method 680
CustomFields.WriteLibrarySearchExtended Method 680
CustomFields.WriteMetadataDefaultForEdit Method 681

CustomFields.WriteMetadataTypeForView Method 681
CustomFields.WritePublicSearch Method 681
CustomFields.WritePublicSearchEx Method 682
Folder Class 683
Folder Methods 683
Folder.AddCommunityFolder Method 684
Folder.AddFolder Method 684
Folder.CreateFolder Method 685
Folder.DeleteFolderById Method 685
GetChildFolders Method 687
Folder.GetChildFolders Method (Long, Boolean, EkEnumeration.FolderOrderBy) 687
Folder.GetChildFolders Method (Long, Boolean, String) 688
GetFolder Method 690
Folder.GetFolder Method (Long) 690
Folder.GetFolder Method (Long, Boolean) 691
Folder.GetFolder Method (Long, Boolean, Boolean) 691
Folder.GetFolderDataWithPermission Method 691
GetFolderId Method 692
Folder.GetFolderId Method (Long) 692
Folder.GetFolderId Method (String, Long) 692
Folder.GetFolderInfo Method 693
Folder.GetPath Method 693
Folder.IsFolderHidden Method 693
Folder.RenameFolder Method 694
UpdateFolder Method 694
Folder.UpdateFolder Method (FolderData) 694
Folder.UpdateFolder Method (FolderRequest) 695
Folder.UpdateFolderApprovalMethod Method 695
Library Class 695
Library Methods 695
Library.AddLibraryItem Method 696
Library.AddLoadBalanceItem Method 696
Library.DeleteLibraryItem Method 697
Library.DeleteLibraryItemById Method 697
Library.DeleteLoadBalanceItem Method 697
Library.GetAllChildLibItems Method 697
GetAllLoadBalancePaths Method 699
Library.GetAllLoadBalancePaths Method (Long, String) 700
Library.GetAllLoadBalancePaths Method (String) 700

GetLibraryItem Method 700
Library.GetLibraryItem Method (Long) 701
Library.GetLibraryItem Method (Long, Long) 701
Library.GetLibrarySettings Method 701
Library.GetLibraryTypes Method 702
Library.GetLoadBalancePath Method 702
Library.InternalLibrarySearch Method 702
Library.UpdateLibrary Method 703
Library.UpdateLibrarySettings Method 703
Library.UpdateLoadBalanceSettings Method 703
Library.UpdateQlinkTemplates Method 703
Localization Class 704
Localization Methods 704
Localization.GetAddViewLanguage Method 705
Localization.GetAddViewLanguageForAllContent Method 705
Localization.GetAddViewLanguageForMenus Method 705
Localization.GetContentItems Method 706
Localization.GetFile Method 706
Localization.GetFiles Method 706
Localization.GetFilesInZipFile Method 707
GetFlagUrl Method 707
Localization.GetFlagUrl Method (Integer) 708
Localization.GetFlagUrl Method (LanguageData) 708
Localization.GetFlagUrl Method (LanguageDataSet.LanguageRow) 708
Localization.GetInterchangeFile Method 709
Localization.GetInterchangeFiles Method 709
Localization.GetInterchangeFilesInZipFile Method 709
GetJobs Method 710
Localization.GetJobs Method () 710
Localization.GetJobs Method (Long) 710
Localization.GetLocalizationUrl Method 711
Localization.GetTranslationUploadDirectory Method 711
Localization.StartExportContentForTranslation Method 711
Localization.StartExportFolderForTranslation Method 712
Localization.StartExportMenusForTranslation Method 712
Localization.StartImportTranslation Method 713
Messaging Class 713
Messaging Methods 713
Messaging.GetUnreadMessageCountForUser Method 713

Metadata Class 714
Metadata Methods 714
AddMetaDataType Method 714
Metadata.AddMetaDataType Method (Collection) 715
Metadata.AddMetaDataType Method (ContentMetaData) 715
Metadata.DeleteMetadataType Method 717
GetContentMetadataList Method 717
Metadata.GetContentMetadataList Method (Long) 717
Metadata.GetContentMetadataList Method (Long, Boolean) 719
Metadata.GetIntegratedSearchFolder Method 719
Metadata.GetIntegratedSearchFolders Method 719
Metadata.GetMetadataByFolder Method 720
Metadata.GetMetadataType Method 720
Metadata.GetMetaDataTypes Method 720
Metadata.UpdateMetadataType Method 721
Permissions Class 721
Permissions Methods 721
Permissions.AddItemPermission Method 722
Permissions.DeleteItemPermission Method 722
Permissions.DisableContentInheritance Method 722
Permissions.DisableFolderInheritance Method 723
Permissions.EnableContentInheritance Method 723
Permissions.EnableFolderInheritance Method 723
GetUserPermissions Method 723
Permissions.GetUserPermissions Method (Long, String, Long, String) 724
Permissions.GetUserPermissions Method (Long, String, Long, String, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) 724
Permissions.LoadPermissions Method 725
Permissions.UpdateItemPermission Method 725
Site Class 725
Site Methods 726
Site.BreakTaskPermissionInherit Method 726
Site.DeleteCookie Method 727
Site.GetAllActiveLanguages Method 727
GetAllLanguages Method 727
Site.GetAllLanguages Method () 727
Site.GetAllLanguages Method (LanguageDataSet.SortBy, LanguageDataSet.SortDirection) 728
Site.GetClientDayNames Method 728
Site.GetClientMonthNames Method 728

Site.GetCookieValue Method 728
Site.GetLanguage Method 729
Site.GetLanguages Method 729
Site.GetSiteVariables Method 729
Site.GetTaskPermission Method 730
Site.IsTaskPermissionInherited Method 730
Site.SetCookieValue Method 730
Site.SetTaskPermission Method 731
Site.UpdateSiteEnabledLanguages Method 731
Site.UpdateSiteVariables Method 731
SiteMap Class 732
SiteMap Methods 732
SiteMap.GetFolderBreadcrumbPath Method 732
SiteMap.GetWebsiteMap Method 734
SiteMap.RenderFolderBreadcrumbHtml Method 735
SiteMap.RenderWebsiteMap Method 736
Subscription Class 736
Subscription Methods 737
Subscription.AddEmailFrom Method 737
Subscription.AddSubscription Method 738
Subscription.AddSubscriptionMessage Method 738
Subscription.DeleteEmailFrom Method 738
Subscription.DeleteSubscription Method 738
Subscription.DeleteSubscriptionMessage Method 739
Subscription.GetAllActiveSubscriptions Method 739
Subscription.GetAllEmailFrom Method 739
Subscription.GetAllSubscriptionMessages Method 739
Subscription.GetAllSubscriptions Method 740
Subscription.GetFolderInheritedFrom Method 740
Subscription.GetSubscription Method 740
Subscription.GetSubscriptionMessage Method 740
Subscription.GetSubscriptionMessagesForType Method 741
Subscription.GetSubscriptionPropertiesForContent Method 741
Subscription.GetSubscriptionPropertiesForFolder Method 741
Subscription.GetSubscriptionsForContent Method 742
Subscription.GetSubscriptionsForFolder Method 742
Subscription.SendWebAlert Method 742
Subscription.UpdateEmailFrom Method 743
Subscription.UpdateSubscription Method 743

Subscription.UpdateSubscriptionMessage Method 743
Subscription.UpdateSubscriptionPropertiesForContent Method 743
Subscription.UpdateSubscriptionPropertiesForFolder Method 744
Utilities Class 744
Utilities Methods 744
Utilities.LoadPermissions Method 745
Ektron.Cms.API.Analytics Namespace 745
Classes 745
Analytics Class 745
Analytics Methods 746
Analytics.GetSessionRank Method 746
Analytics.HardDumpAnalyticsBuffer Method 746
Analytics.QueryAnalytics Method 746
Analytics.SoftDumpAnalyticsBuffer Method 747
TrackVisit Method 747
Analytics Properties 748
Analytics.AnalyticsEnabled Property 748
Analytics.AppendToIISLog Property 749
Analytics.InMemoryContentHitsData Property 749
Analytics.RecordsBeforeWrite Property 749
Analytics.TimeBeforeWrite Property 749
Ektron.Cms.API.Community Namespace 750
Classes 750
CommunityGroup Class 750
CommunityGroup Methods 751
CommunityGroup.AcceptGroupInvite Method 752
CommunityGroup.AddCommunityGroup Method 752
AddUserToCommunityGroup Method 752
CommunityGroup.ApprovePendingGroupUser Method 753
CommunityGroup.CancelJoinRequestForCommunityGroup Method 754
CommunityGroup.DeleteCommunityGroupByID Method 754
CommunityGroup.DeletePendingGroupUser Method 754
CommunityGroup.GetAllCommunityGroups Method 754
CommunityGroup.GetAllCommunityGroupsInDirectory Method 755
CommunityGroup.GetAllUnassignedCommunityGroupUsers Method 755
CommunityGroup.GetCommunityGroupBlog Method 755
CommunityGroup.GetCommunityGroupBlogIds Method 756
CommunityGroup.GetCommunityGroupByID Method 756
CommunityGroup.GetCommunityGroupDiscussionBoardIds Method 756

CommunityGroup.GetCommunityGroups Method 757
CommunityGroup.GetCommunityGroupUserIDs Method 757
GetCommunityGroupUsers Method 757
CommunityGroup.GetFolderIdForCommunityGroup Method 758
CommunityGroup.GetGroupMemberStatus Method 759
CommunityGroup.GetInvitedCommunityGroups Method 759
CommunityGroup.GetList Method 759
GetMostPopularCommunityGroups Method 760
CommunityGroup.GetPendingCommunityGroups Method 761
CommunityGroup.GetPendingCommunityGroupUsers Method 761
CommunityGroup.InviteUsers Method 761
CommunityGroup.IsCommunityGroupAdmin Method 762
CommunityGroup.IsUserInCommunityGroup Method 762
CommunityGroup.RemoveUserFromCommunityGroup Method 762
CommunityGroup.UpdateCommunityGroup Method 763
Favorites Class 763
Favorites Methods 763
AddContentFavorite Method 764
Favorites.AddFavoriteLink Method 765
Favorites.DeleteContentFavorite Method 765
Favorites.DeleteFavoriteLink Method 765
GetAllFavoriteLink Method 766
GetContentFavorites Method 767
Favorites.GetFavoriteId Method 769
Favorites.GetFavoriteLink Method 769
Favorites.GetFavoritesDirectory Method 770
Favorites.IsContentFavorite Method 770
Favorites.MoveContentFavorite Method 770
Favorites.MoveFavoriteLink Method 771
Favorites.UpdateFavoriteLink Method 771
Flagging Class 771
Flagging Methods 772
Flagging.AddFlagObject Method 772
Flagging.AssignFlagToContent Method 772
GetAllFlagEntries Method 773
GetContentFlagData Method 774
Flagging.GetDefaultFolderFlagDef Method 775
Flagging.GetFlagDataById Method 775
Flagging.GetFlaggedRank Method 776

Flagging.PurgeFlagEntries Method 776
Flagging.RemoveObjectFlag Method 776
Flagging.UpdateFlagObject Method 777
Friends Class 777
Friends Methods 777
Friends.AcceptFriendInvite Method 778
Friends.AcceptPendingFriend Method 778
AddFriend Method 778
Friends.AddPendingFriend Method 779
Friends.AddSelectedFriend Method 780
Friends.DeleteFriend Method 780
DeletePendingFriendRequest Method 780
Friends.DeleteSelectedFriend Method 781
GetFriends Method 781
Friends.GetFriendsDirectory Method 783
Friends.GetFriendStatus Method 784
Friends.GetInvitedUsers Method 784
Friends.GetPendingFriends Method 784
Friends.GetSelectedFriends Method 785
Friends.GetSentFriendInvitations Method 785
Friends.InviteUsers Method 785
Friends.IsFriend Method 786
Friends.IsSelectedFriend Method 786
Friends.MoveFriend Method 786
MessageBoard Class 787
MessageBoard Methods 787
MessageBoard.AddMessageBoardEntry Method 788
ApproveMessage Method 788
DeleteMessageBoardEntries Method 790
DeleteMessageBoardEntriesByUser Method 792
DeleteMessageBoardEntry Method 793
MessageBoard.DisableModeration Method 795
MessageBoard.EnableModeration Method 795
GetAllMessageBoardEntries Method 796
GetAllMessageBoardEntriesByUser Method 798
GetMessageBoardEntry Method 800
MessageBoard.IsModerated Method 802
MessageBoard.IsSpam Method 802
MessageBoard.IsUserMessageBoardAdmin Method 802

UpdateMessageText Method 803
Tags Class 805
Tags Methods 805
Tags.AddTag Method 806
AddTagToCommunityGroup Method 806
AddTagToContent Method 808
AddTagToObject Method 809
AddTagToUser Method 811
Tags.DeleteTagById Method 812
Tags.DeleteTagOnObject Method 813
Tags.DeleteTagsByUser Method 813
Tags.DeleteTagsOnCommunityGroup Method 813
Tags.DeleteTagsOnContent Method 814
Tags.DeleteTagsOnObject Method 814
Tags.DeleteTagsOnUser Method 814
Tags.EditTag Method 814
Tags.GetAllTagRelatedContent Method 815
Tags.GetAllTagRelatedUsers Method 815
GetAllTags Method 815
GetDefaultTags Method 816
Tags.GetTagByID Method 818
Tags.GetTagByName Method 818
Tags.GetTagCloud Method 818
Tags.GetTagsForContent Method 818
GetTagsForObject Method 819
GetTagsForUser Method 820
Tags.RemoveTagAsDefault Method 821
Tags.SaveTagAsDefault Method 821
Ektron.Cms.API.Content Namespace 822
Classes 822
Asset Class 822
Asset Methods 823
Asset.GetAssetMgtConfigInfo Method 823
Asset.GetAssetMgtInfo Method 823
Asset.GetAssetSupertypes Method 824
Asset.GetEditSnippet Method 824
Asset.GetMyAssetMgtID Method 824
Asset.GetViewUrl Method 825
Asset.MakeCopyDMSAsset Method 825

Asset.SetAssetMgtConfigInfo Method 825
Blog Class 826
Blog Methods 826
Blog.AddBlog Method 827
Blog.AddCommentForPost Method 827
Blog.AddPost Method 827
Blog.DeleteBlog Method 829
Blog.DeleteBlogPostComment Method 830
Blog.DeletePost Method 830
Blog.GetBlankBlogPostData Method 830
GetBlog Method 830
GetBlogData Method 834
Blog.GetBlogDataForPost Method 837
Blog.GetBlogPostData Method 837
GetBlogRoll Method 837
Blog.GetPost Method 839
Blog.GetPostbyID Method 839
Blog.GetUserBlog Method 839
Blog.UpdateBlogProperties Method 840
Content Class 840
Content Methods 840
Content.AddAsset Method 842
AddContent Method 842
Content.AddXmlConfiguration Method 847
Content.CheckOutContent Method 847
Content.CopyContentToFolder Method 847
Content.DeleteContentItem Method 848
Content.DeleteXmlConfiguration Method 849
Content.EditAsset Method 849
Content.GetAddViewLanguage Method 849
Content.GetAllComments Method 850
Content.GetAllTemplates Method 850
Content.GetAllUnassignedItemApprovals Method 850
GetAllXmlConfigurations Method 851
GetChildContent Method 852
Content.GetCommunityTemplate Method 854
GetContent Method 854
Content.GetContentByHistory Method 857
Content.GetContentDisplayLink Method 858

Content.GetContentForEditing Method 858
GetContentReport Method 858
Content.GetContentState Method 859
Content.GetContentStatus Method 860
Content.GetDomain Method 860
Content.GetExpireContent Method 860
GetList Method 860
Content.MoveContentToFolder Method 861
Content.PublishContent Method 862
Content.RelicateXmlConfiguration Method 862
Content.RenameContent Method 863
Content.ReplicateXmlConfiguration Method 863
Content.RestoreHistoryContent Method 863
Content.SaveContent Method 863
Content.SubmitForDelete Method 864
Content.UndoCheckout Method 864
Content.UpdateContentMetaData Method 865
Content.UpdatexmlCollectionPackage Method 865
Content.UpdateXmlConfiguration Method 865
ContentRating Class 866
ContentRating Methods 866
ContentRating.GetContentRating Method 866
ContentRating.GetContentRatingAddRating Method 868
ContentRating.GetContentRatingResults Method 869
GetContentRatingStatistics Method 869
ContentRating.PurgeContentRatings Method 870
Form Class 871
Form Methods 871
Form.AssignFormTask Method 871
Form.EmailFormFieldData Method 872
GetAllForms Method 872
Form.GetForm Method 873
GetFormFieldData Method 873
Form.GetFormFieldList Method 874
Form.GetFormFieldQuestions Method 875
Form.GetFormTitle Method 875
Form.SubmitFormFieldData Method 875
Form.UpdateFormFieldData Method 876
Taxonomy Class 876

Taxonomy Methods 876
AddTaxonomyItem Method 877
Taxonomy.AddTaxonomySyncFolder Method 878
Taxonomy.CloneTaxonomy Method 878
Taxonomy.CreateTaxonomy Method 879
Taxonomy.DeleteTaxonomy Method 879
Taxonomy.GetTaxonomyIdByPath Method 879
ImportTaxonomy Method 880
Taxonomy.LoadTaxonomy Method 880
Taxonomy.ReadAllAssignedCategory Method 881
Taxonomy.ReadTaxonomy Method 881
Taxonomy.RemoveTaxonomyFolder Method 881
Taxonomy.RemoveTaxonomyItem Method 882
Taxonomy.ReOrderTaxonomyItems Method 882
Taxonomy.UpdateTaxonomy Method 882
ThreadedDiscussion Class 882
ThreadedDiscussion Methods 883
ThreadedDiscussion.AddBoard Method 884
ThreadedDiscussion.AddForum Method 884
AddReply Method 886
AddTopic Method 888
ThreadedDiscussion.DeleteBoard Method 889
ThreadedDiscussion.DeleteForum Method 889
ThreadedDiscussion.DeleteTopic Method 889
GetActiveTopics Method 890
ThreadedDiscussion.GetActiveTopicsForForum Method 891
ThreadedDiscussion.GetActiveUsers Method 891
ThreadedDiscussion.GetBoardCategories Method 891
ThreadedDiscussion.GetBoardForums Method 892
GetDiscussionBoard Method 892
ThreadedDiscussion.GetDiscussionBoardString Method 893
ThreadedDiscussion.GetForum Method 893
ThreadedDiscussion.GetRepliesForTopics Method 895
ThreadedDiscussion.GetTopic Method 895
ThreadedDiscussion.SearchReplies Method 896
ThreadedDiscussion.SubscribeToBoard Method 896
ThreadedDiscussion.SubscribeToForum Method 897
ThreadedDiscussion.SubscribeToTopic Method 897
ThreadedDiscussion.UnSubscribeToBoard Method 897

ThreadedDiscussion.UnSubscribeToForum Method 898
ThreadedDiscussion.UnSubscribeToTopic Method 898
Ektron.Cms.API.Search Namespace 899
Classes 899
SearchManager Class 899
SearchManager Methods 900
Execute Method 900
Search Method 902
SearchForCategories Method 906
SearchTable Method 907
Ektron.Cms.API.Task Namespace 909
Classes 909
Task Class 909
Task Methods 909
Task.AddTask Method 910
AssignTaskToUser Method 910
AssignTaskToUsergroup Method 911
DeleteTask Method 912
DeleteTasks Method 913
Task.GetTask Method 914
GetTasks Method 914
GetTasksAssignedByUser Method 915
GetTasksAssignedToUser Method 916
GetTasksAssignedToUsergroup Method 917
Task.GetTaskState Method 918
Task.SetStateForTask Method 919
Task.UpdateTask Method 919
TaskCategory Class 919
TaskCategory Methods 920
TaskCategory.AddTaskCategory Method 920
TaskCategory.DeleteTaskCategory Method 920
TaskCategory.GetAllCategories Method 921
TaskCategory.GetTaskCategory Method 921
TaskCategory.UpdateTaskCategory Method 921
TaskCategoryType Class 922
TaskCategoryType Methods 922
TaskCategoryType.AddTaskType Method 922
TaskCategoryType.DeleteTaskType Method 923
TaskCategoryType.GetTaskType Method 923

TaskCategoryType.UpdateTaskType Method 923
Ektron.Cms.API.UrlAliasing Namespace 924
Classes 924
UrlAliasAuto Class 924
UrlAliasAuto Methods 924
UrlAliasAuto.Add Method 925
UrlAliasAuto.ClearCache Method 925
UrlAliasAuto.Delete Method 926
UrlAliasAuto.GetAutoAliasNameTypes Method 926
UrlAliasAuto.GetAutoAliasTypes Method 926
UrlAliasAuto.GetFolderItem Method 926
UrlAliasAuto.GetFolderList Method 927
UrlAliasAuto.GetFolderSource Method 927
UrlAliasAuto.GetItem Method 927
UrlAliasAuto.GetItemForContent Method 928
UrlAliasAuto.GetList Method 928
UrlAliasAuto.GetListForContent Method 929
GetTarget Method 929
UrlAliasAuto.GetTaxonomyItem Method 930
UrlAliasAuto.GetTaxonomyList Method 931
UrlAliasAuto.GetTaxonomySource Method 931
UrlAliasAuto.Update Method 931
UrlAliasCommon Class 931
UrlAliasCommon Methods 932
UrlAliasCommon.GetAliasForContent Method 932
GetTarget Method 932
UrlAliasManual Class 933
UrlAliasManual Methods 933
UrlAliasManual.Add Method 934
UrlAliasManual.ClearCache Method 934
UrlAliasManual.Delete Method 934
UrlAliasManual.GetDefaultAlias Method 934
UrlAliasManual.GetFileExtensions Method 935
UrlAliasManual.GetItem Method 935
GetList Method 935
UrlAliasManual.GetSiteList Method 940
UrlAliasManual.GetTemplateList Method 940
UrlAliasManual.Update Method 940
UrlAliasManual.ValidateName Method 940

UrlAliasRegEx Class 941
UrlAliasRegEx Methods 941
UrlAliasRegEx.Add Method 941
UrlAliasRegEx.ClearCache Method 942
UrlAliasRegEx.Delete Method 942
UrlAliasRegEx.EvaluateExpression Method 942
UrlAliasRegEx.GetItem Method 942
UrlAliasRegEx.GetList Method 943
UrlAliasRegEx.GetPriorityList Method 943
UrlAliasRegEx.GetTarget Method 944
UrlAliasRegEx.Update Method 944
UrlAliasSettings Class 944
UrlAliasSettings Methods 945
UrlAliasSettings.RefreshSettingsCache Method 945
UrlAliasSettings Properties 945
UrlAliasSettings.AutoCacheDuration Property 946
UrlAliasSettings.AutoCacheSize Property 946
UrlAliasSettings.DefaultPage Property 947
UrlAliasSettings.DisableLanguageAwareness Property 947
UrlAliasSettings.EnableAutoAliasing Property 947
UrlAliasSettings.EnableAutoCaching Property 947
UrlAliasSettings.EnabledTracing Property 947
UrlAliasSettings.EnableManualAliasing Property 948
UrlAliasSettings.EnableManualCaching Property 948
UrlAliasSettings.EnableRegExAliasing Property 948
UrlAliasSettings.EnableRegExCaching Property 948
UrlAliasSettings.IsAutoAliasEnabled Property 949
UrlAliasSettings.IsManualAliasEnabled Property 949
UrlAliasSettings.IsRegExAliasEnabled Property 949
UrlAliasSettings.ManualCacheDuration Property 949
UrlAliasSettings.ManualCacheSize Property 949
UrlAliasSettings.OverrideTemplate Property 950
UrlAliasSettings.PageExtensions Property 950
UrlAliasSettings.QueryStringAction Property 950
UrlAliasSettings.RegExCacheDuration Property 950
UrlAliasSettings.RegExCacheSize Property 950
Ektron.Cms.API.User Namespace 950
Classes 951
ADUser Class 951

ADUser Methods 951
ADUser.AddADGroupToCMS Method 952
ADUser.AddADUsersToCMSByUsername Method 952
ADUser.DeSynchUserGroups Method 952
ADUser.DeSynchUsers Method 953
ADUser.GetADMapping Method 953
ADUser.GetADStatus Method 953
ADUser.GetAvailableADGroups Method 954
ADUser.GetAvailableADUsers Method 954
ADUser.GetCMSGroupsToSync Method 954
ADUser.GetCMSRelationshipsToSync Method 955
ADUser.GetCMSUsersToSync Method 955
ADUser.GetDomains Method 955
ADUser.MapCMSUserGroupToAD Method 956
ADUser.MapCMSUserToAD Method 956
ADUser.SynchCMSGroupsToAD Method 956
ADUser.SynchCMSRelationshipsToAD Method 957
ADUser.SynchCMSUsersToAD Method 957
ADUser.UpdateADMapping Method 957
ADUser.UpdateADUsersGroups Method 958
ADUser.UpdateUsersInfoFromAD Method 958
User Class 958
User Methods 958
ActivateUserAccount Method 960
User.AddMembershipUser Method 961
User.AddNewUser Method 963
User.AddRolePermission Method 964
AddUser Method 964
User.AddUserGroup Method 965
User.AddUserToGroup Method 965
User.AutoLogInUser Method 965
CreateBlankUserObject Method 966
User.DeleteUserByID Method 967
User.DeleteUserByIds Method 967
User.DeleteUserFromGroup Method 967
User.DeleteUserGroup Method 967
User.EditUserCustomProperties Method 968
User.GetActiveUser Method 968
User.GetADMapping Method 968

User.GetAllCmsUsers Method 969
User.GetAllCustomProperties Method 969
User.GetAllUserByCustomProperty Method 970
GetAllUserGroups Method 970
GetAllUsers Method 971
User.GetGroupsUserIsIn Method 972
User.GetMembershipUser Method 972
GetNonVerifyUser Method 973
User.GetUser Method 973
User.GetUserByUsername Method 974
User.GetUserCustomProperties Method 975
User.GetUserEmailInfo Method 976
User.GetUserForFolderAdmin Method 976
User.GetUserGroup Method 976
User.GetUserGroupByName Method 976
User.GetUserGroupForFolderAdmin Method 977
User.GetUserPermissions Method 977
User.GetUserPreference Method 978
GetUsers Method 978
User.GetUsersForTask Method 979
User.GetUsersNotInGroup Method 980
User.IsAdmin Method 980
User.IsAGroupMember Method 980
IsARoleMember Method 981
User.IsARoleMemberForContent Method 984
User.IsARoleMemberForContent_FolderUserAdmin Method 984
User.IsARoleMemberForFolder Method 984
User.IsARoleMemberForFolder_FolderUserAdmin Method 985
User.IsARoleMemberForFolder_MetadataAdmin Method 985
User.LockUserAccount Method 986
LogInUser Method 986
User.ReadCustomProperties Method 990
ResetMembershipUserPassword Method 990
User.SetAuthenticationCookie Method 991
User.UnsubscribeUser Method 991
User.UpdateCustomPropertiesItemOrder Method 992
UpdateUser Method 992
UpDateUserGroup Method 993
User.UpdateUserPreferences Method 993

UserObject Method 994
User Properties 995
User.UserIdAuthenticated Property 995
Ektron.Cms.Framework Namespace 995
Ektron.Cms.Framework.Activity Namespace 996
Classes 996
Activity Class 996
Activity.Activity Constructor 996
Activity Methods 997
Activity.Add Method 997
Activity.AddActivityMessage Method 997
Activity.Delete Method 998
Activity.DisableActivityPublishing Method 998
Activity.EnableActivityPublishing Method 998
Activity.GetItem Method 998
GetList Method 998
Activity.GetListForUser Method 999
Activity.Publish Method 1000
Activity.Update Method 1000
Activity.UpdateActivityMessage Method 1000
Activity Properties 1000
Activity.IsActivityPublishingEnabled Property 1001
ActivityQueue Class 1001
ActivityStream Class 1001
ActivityStream.ActivityStream Constructor 1001
ActivityStream Methods 1002
GetUserActivityStream Method 1002
ActivityStream.SendActivityToAllStreams Method 1003
ActivityStream.SendActivityToStreams Method 1003
ActivityStream.SendActivityToUsersStream Method 1003
ActivityType Class 1004
ActivityType Methods 1004
ActivityType.Add Method 1004
ActivityType.Delete Method 1004
ActivityType.GetItem Method 1005
ActivityType.GetList Method 1005
ActivityType.Update Method 1005
Ektron.Cms.Framework.Analytics Namespace 1005
Classes 1006

ReportAnalytics Class 1006
ReportAnalytics.ReportAnalytics Constructor 1006
ReportAnalytics Methods 1006
GetAbsoluteUniqueVisitorsTrend Method 1007
GetAdVersions Method 1009
GetAveragePageViewsTrend Method 1010
GetBounceRateTrend Method 1011
GetBrowsers Method 1012
GetCampaigns Method 1013
ReportAnalytics.GetCmsSearchTerms Method 1014
GetConnectionSpeeds Method 1014
GetContentbyTitle Method 1015
GetContentDetail Method 1016
GetFlashVersions Method 1017
GetHostnames Method 1018
GetJavaSupport Method 1020
GetKeywords Method 1021
GetLanguages Method 1022
GetLocations Method 1023
GetNetworkLocations Method 1024
GetNewVsReturningVisitors Method 1025
GetOperatingSystems Method 1026
GetPageViewsTrend Method 1027
GetPlatforms Method 1028
GetScreenColors Method 1029
GetScreenResolutions Method 1030
GetTimeOnSiteTrend Method 1031
GetTopContent Method 1032
GetTopExitPages Method 1033
GetTopLandingPages Method 1034
GetVisitsTrend Method 1035
SearchAnalytics Class 1036
SearchAnalytics.SearchAnalytics Constructor 1037
Ektron.Cms.Framework.Calendar Namespace 1037
Classes 1037
WebCalendar Class 1037
WebCalendar Constructor 1038
WebCalendar.WebCalendar Constructor () 1038
WebCalendar.WebCalendar Constructor (ApiAccessMode) 1038

WebCalendar Methods 1038
WebCalendar.Add Method 1039
WebCalendar.Delete Method 1039
WebCalendar.GetCalendar Method 1039
WebCalendar.GetICalendarFeed Method 1039
WebCalendar.GetPublicCalendar Method 1039
WebCalendar.Update Method 1040
WebEvent Class 1040
WebEvent Constructor 1040
WebEvent.WebEvent Constructor () 1041
WebEvent.WebEvent Constructor (ApiAccessMode) 1041
WebEvent Methods 1041
WebEvent.Add Method 1041
WebEvent.CancelOccurrence Method 1042
CreateVariance Method 1042
WebEvent.Delete Method 1043
WebEvent.GetEventOccurrenceList Method 1043
WebEvent.GetItem Method 1044
GetList Method 1044
GetNonVariantEventList Method 1045
GetVarianceEventList Method 1046
WebEvent.Update Method 1047
Ektron.Cms.Framework.Core Namespace 1047
Ektron.Cms.Framework.Core.Content Namespace 1047
Classes 1047
Content Class 1048
Content.Content Constructor 1048
Content Methods 1048
Ektron.Cms.Framework.Core.Folder Namespace 1050
Classes 1050
Folder Class 1050
Folder.Folder Constructor 1050
Folder Methods 1050
Ektron.Cms.Framework.Messaging Namespace 1052
Classes 1052
CmsMessage Class 1052
CmsMessage Constructor 1053
CmsMessage.CmsMessage Constructor () 1053
CmsMessage.CmsMessage Constructor (ApiAccessMode) 1053

CmsMessage Fields 1053
CmsMessage._messageService Field 1053
CmsMessage Methods 1054
CmsMessage.Add Method 1054
CmsMessage.Delete Method 1054
CmsMessage.GetDefaultItemByType Method 1054
CmsMessage.GetItem Method 1055
CmsMessage.GetItemCollection Method 1055
CmsMessage.GetList Method 1055
CmsMessage.Update Method 1055
CmsMessage Properties 1056
CmsMessage.MessageService Property 1056
CmsMessageType Class 1056
CmsMessageType Constructor 1056
CmsMessageType.CmsMessageType Constructor () 1057
CmsMessageType.CmsMessageType Constructor (ApiAccessMode) 1057
CmsMessageType Fields 1057
CmsMessageType._messageTypeService Field 1057
CmsMessageType Methods 1057
CmsMessageType.Add Method 1058
CmsMessageType.Delete Method 1058
CmsMessageType.GetItem Method 1058
CmsMessageType.GetList Method 1058
CmsMessageType.GetTokenList Method 1059
CmsMessageType.Update Method 1059
CmsMessageType Properties 1059
CmsMessageType.MessageTypeService Property 1059
Ektron.Cms.Framework.Notifications Namespace 1060
Classes 1060
Notification Class 1060
Notification.Notification Constructor 1060
Notification Methods 1061
Send Method 1061
NotificationAgentSetting Class 1062
NotificationAgentSetting.NotificationAgentSetting Constructor 1062
NotificationAgentSetting Methods 1062
NotificationAgentSetting.Add Method 1062
NotificationAgentSetting.Delete Method 1063
NotificationAgentSetting.GetAgent Method 1063

NotificationAgentSetting.GetItem Method 1063
NotificationAgentSetting.GetList Method 1063
NotificationAgentSetting.GetRegisteredAgentList Method 1064
NotificationAgentSetting.Update Method 1064
NotificationPreference Class 1064
NotificationPreference.NotificationPreference Constructor 1064
NotificationPreference Methods 1065
NotificationPreference.Add Method 1065
NotificationPreference.Delete Method 1065
NotificationPreference.GetDefaultPreferenceList Method 1065
NotificationPreference.GetItem Method 1066
NotificationPreference.GetList Method 1066
NotificationPreference.GetUserNotificationListForActivity Method 1066
NotificationPreference.SaveUserPreferences Method 1066
NotificationPreference.Update Method 1067
UserNotificationSetting Class 1067
UserNotificationSetting.UserNotificationSetting Constructor 1067
UserNotificationSetting Methods 1068
UserNotificationSetting.Add Method 1068
UserNotificationSetting.Delete Method 1068
UserNotificationSetting.GetItem Method 1068
UserNotificationSetting.GetList Method 1069
UserNotificationSetting.Update Method 1069
UserNotificationSetting.VerifyValidationCode Method 1069
Ektron.Cms.Framework.SocialNetworking Namespace 1069
Classes 1070
MicroMessage Class 1070
MicroMessage Constructor 1070
MicroMessage.MicroMessage Constructor () 1070
MicroMessage.MicroMessage Constructor (ApiAccessMode) 1071
MicroMessage Fields 1071
MicroMessage._MicroMessageBL Field 1071
MicroMessage Methods 1071
MicroMessage.Add Method 1072
MicroMessage.AddReply Method 1072
MicroMessage.Delete Method 1072
GetColleagueMessageList Method 1072
MicroMessage.GetItem Method 1073
MicroMessage.GetList Method 1073

MicroMessage.GetReplyList Method 1074
MicroMessage.GetSearchList Method 1074
GetTimeLineMessageList Method 1074
MicroMessage.GetUserMessageList Method 1075
MicroMessage.IsFullTextSearchInstalled Method 1075
MicroMessage.IsSpam Method 1075
MicroMessage Properties 1076
MicroMessage.GetInstance Property 1076
Ektron.Cms.Framework.Users Namespace 1076
Classes 1076
User Class 1076
User Constructor 1077
User.User Constructor () 1077
User.User Constructor (ApiAccessMode) 1077
User Fields 1077
User._userService Field 1078
User Methods 1078
User.Add Method 1078
User.Delete Method 1078
User.GetItem Method 1078
User.GetList Method 1079
User.Update Method 1079
User Properties 1079
User.UserService Property 1079
Classes 1080
CmsApi Class 1080
CmsApi Constructor 1080
CmsApi.CmsApi Constructor () 1080
CmsApi.CmsApi Constructor (ApiAccessMode) 1081
CmsApi Properties 1081
CmsApi.ApiMode Property 1081
CmsApi.ContentLanguage Property 1081
CmsApi.InPreviewMode Property 1081
CmsApi.RequestInformation Property 1082
CmsApi.SitePath Property 1082
CmsApi.UserId Property 1082
Structs, Records, Enums 1082
Ektron.Cms.Framework.ApiAccessMode Enumeration 1082

Web Services API **1083**

Ektron.Services Namespace 1083
Ektron.Services.Task Namespace 1085
Classes 1085
Task Class 1085
Task Methods 1085
Task.AddTask Method 1086
AssignTaskToUser Method 1086
AssignTaskToUsergroup Method 1087
DeleteTask Method 1089
DeleteTasks Method 1090
Task.GetTask Method 1091
Task.GetTasks Method 1091
GetTasksAssignedByUser Method 1092
GetTasksAssignedToUser Method 1093
GetTasksAssignedToUsergroup Method 1094
Task.GetTaskState Method 1095
Task.SetStateForTask Method 1095
Task.UpdateTask Method 1096
TaskCategory Class 1096
TaskCategory Methods 1097
TaskCategory.AddTaskCategory Method 1097
TaskCategory.DeleteTaskCategory Method 1097
TaskCategory.GetAllCategories Method 1098
TaskCategory.GetTaskCategory Method 1098
TaskCategory.UpdateTaskCategory Method 1099
TaskCategoryType Class 1099
TaskCategoryType Methods 1099
TaskCategoryType.AddTaskType Method 1100
TaskCategoryType.DeleteTaskType Method 1100
TaskCategoryType.GetTaskType Method 1100
TaskCategoryType.UpdateTaskType Method 1101
Classes 1101
ActiveDirectoryUser Class 1103
ActiveDirectoryUser Methods 1103
ActiveDirectoryUser.AddADGroupToCMS Method 1104
ActiveDirectoryUser.DeSynchUserGroups Method 1104
ActiveDirectoryUser.DeSynchUsers Method 1105
ActiveDirectoryUser.GetADMapping Method 1105
ActiveDirectoryUser.GetADStatus Method 1106

ActiveDirectoryUser.GetAvailableADGroups Method 1106
ActiveDirectoryUser.GetAvailableADUsers Method 1107
ActiveDirectoryUser.GetCMSGroupsToSync Method 1108
ActiveDirectoryUser.GetCMSRelationshipsToSync Method 1108
ActiveDirectoryUser.GetCMSUsersToSync Method 1109
ActiveDirectoryUser.GetDomains Method 1109
ActiveDirectoryUser.MapCMSUserGroupToAD Method 1110
ActiveDirectoryUser.MapCMSUserToAD Method 1110
ActiveDirectoryUser.UpdateADMapping Method 1111
ActiveDirectoryUser.UpdateADUsersGroups Method 1111
ActiveDirectoryUser.UpdateUsersInfoFromAD Method 1112
Asset Class 1112
Asset Methods 1113
Asset.GetAssetMgtConfigInfo Method 1113
Asset.GetAssetMgtInfo Method 1113
Asset.GetAssetSupertypes Method 1114
Asset.GetEditSnippet Method 1114
Asset.GetMyAssetMgtID Method 1115
Asset.MakeCopyDMSAsset Method 1115
Asset.SetAssetMgtConfigInfo Method 1115
Blog Class 1116
Blog Methods 1117
Blog.AddBlog Method 1117
Blog.AddCommentForPost Method 1118
Blog.AddPost Method 1119
Blog.DeleteBlog Method 1119
Blog.DeleteBlogPostComment Method 1120
Blog.DeletePost Method 1120
Blog.GetBlankBlogPostData Method 1120
GetBlog Method 1121
Blog.GetBlog Method (long, string, int, int) 1121
Blog.GetBlog Method (string, string, int, int) 1122
GetBlogData Method 1123
Blog.GetBlogData Method (FolderData) 1123
Blog.GetBlogData Method (long) 1124
Blog.GetBlogData Method (long, DateTime) 1125
Blog.GetBlogData Method (long, DateTime, int, int) 1125
Blog.GetBlogDataForPost Method 1126
Blog.GetBlogPostData Method 1127

GetBlogRoll Method 1127
Blog.GetBlogRoll Method (long) 1128
Blog.GetBlogRoll Method (string) 1128
Blog.GetPost Method 1129
Blog.GetPostbyID Method 1129
Blog.GetUserBlog Method 1130
Blog.UpdateBlogProperties Method 1130
BusinessRules Class 1130
BusinessRules Methods 1131
GetBusinessRules Method 1131
BusinessRules.GetBusinessRules Method (string, string, string, string, string, string, string, long) 1132
BusinessRules.GetBusinessRules Method (string, string, string, string, string, string, string, string) 1132
Calendar Class 1133
Analytics Class 1133
Analytics Methods 1134
GetAbsoluteUniqueVisitorsTrend Method 1135
Analytics.GetAbsoluteUniqueVisitorsTrend Method (DateTime, DateTime, AnalyticsCriteria) 1135
Analytics.GetAbsoluteUniqueVisitorsTrend Method (string, DateTime, DateTime, AnalyticsCriteria) 1136
GetAdVersions Method 1136
Analytics.GetAdVersions Method (DateTime, DateTime, AnalyticsCriteria) 1137
Analytics.GetAdVersions Method (string, DateTime, DateTime, AnalyticsCriteria) 1137
GetAllTrafficSources Method 1138
Analytics.GetAllTrafficSources Method (DateTime, DateTime, AnalyticsCriteria) 1138
Analytics.GetAllTrafficSources Method (string, DateTime, DateTime, AnalyticsCriteria) 1139
GetAveragePageViewsTrend Method 1139
Analytics.GetAveragePageViewsTrend Method (DateTime, DateTime, AnalyticsCriteria) 1140
Analytics.GetAveragePageViewsTrend Method (string, DateTime, DateTime, AnalyticsCriteria) 1140
GetBounceRateTrend Method 1141
Analytics.GetBounceRateTrend Method (DateTime, DateTime, AnalyticsCriteria) 1141
Analytics.GetBounceRateTrend Method (string, DateTime, DateTime, AnalyticsCriteria) 1141
GetBrowsers Method 1142
Analytics.GetBrowsers Method (DateTime, DateTime, AnalyticsCriteria) 1142
Analytics.GetBrowsers Method (string, DateTime, DateTime, AnalyticsCriteria) 1143
GetCampaigns Method 1143
Analytics.GetCampaigns Method (DateTime, DateTime, AnalyticsCriteria) 1144
Analytics.GetCampaigns Method (string, DateTime, DateTime, AnalyticsCriteria) 1144

Analytics.GetCmsSearchTerms Method 1145
GetConnectionSpeeds Method 1145
Analytics.GetConnectionSpeeds Method (DateTime, DateTime, AnalyticsCriteria) 1146
Analytics.GetConnectionSpeeds Method (string, DateTime, DateTime, AnalyticsCriteria) 1146
GetContentbyTitle Method 1147
Analytics.GetContentbyTitle Method (DateTime, DateTime, AnalyticsCriteria) 1147
Analytics.GetContentbyTitle Method (string, DateTime, DateTime, AnalyticsCriteria) 1148
GetContentDetail Method 1148
Analytics.GetContentDetail Method (DateTime, DateTime, AnalyticsCriteria) 1149
Analytics.GetContentDetail Method (string, DateTime, DateTime, AnalyticsCriteria) 1149
GetDirectTraffic Method 1150
Analytics.GetDirectTraffic Method (DateTime, DateTime, AnalyticsCriteria) 1150
Analytics.GetDirectTraffic Method (string, DateTime, DateTime, AnalyticsCriteria) 1151
GetFlashVersions Method 1151
Analytics.GetFlashVersions Method (DateTime, DateTime, AnalyticsCriteria) 1151
Analytics.GetFlashVersions Method (string, DateTime, DateTime, AnalyticsCriteria) 1152
GetHostnames Method 1153
Analytics.GetHostnames Method (DateTime, DateTime, AnalyticsCriteria) 1153
Analytics.GetHostnames Method (string, DateTime, DateTime, AnalyticsCriteria) 1153
GetJavaSupport Method 1154
Analytics.GetJavaSupport Method (DateTime, DateTime, AnalyticsCriteria) 1154
Analytics.GetJavaSupport Method (string, DateTime, DateTime, AnalyticsCriteria) 1155
GetKeywords Method 1155
Analytics.GetKeywords Method (DateTime, DateTime, AnalyticsCriteria) 1156
Analytics.GetKeywords Method (string, DateTime, DateTime, AnalyticsCriteria) 1156
GetLanguages Method 1157
Analytics.GetLanguages Method (DateTime, DateTime, AnalyticsCriteria) 1157
Analytics.GetLanguages Method (string, DateTime, DateTime, AnalyticsCriteria) 1158
GetLocations Method 1158
Analytics.GetLocations Method (DateTime, DateTime, AnalyticsCriteria) 1159
Analytics.GetLocations Method (string, DateTime, DateTime, AnalyticsCriteria) 1159
GetNetworkLocations Method 1160
Analytics.GetNetworkLocations Method (DateTime, DateTime, AnalyticsCriteria) 1160
Analytics.GetNetworkLocations Method (string, DateTime, DateTime, AnalyticsCriteria) 1161
GetNewVsReturningVisitors Method 1161
Analytics.GetNewVsReturningVisitors Method (DateTime, DateTime, AnalyticsCriteria) 1161
Analytics.GetNewVsReturningVisitors Method (string, DateTime, DateTime, AnalyticsCriteria) 1162
GetOperatingSystems Method 1163

Analytics.GetOperatingSystems Method (DateTime, DateTime, AnalyticsCriteria) 1163
Analytics.GetOperatingSystems Method (string, DateTime, DateTime, AnalyticsCriteria) 1163
GetPageViewsTrend Method 1164
Analytics.GetPageViewsTrend Method (DateTime, DateTime, AnalyticsCriteria) 1164
Analytics.GetPageViewsTrend Method (string, DateTime, DateTime, AnalyticsCriteria) 1165
GetPlatforms Method 1165
Analytics.GetPlatforms Method (DateTime, DateTime, AnalyticsCriteria) 1166
Analytics.GetPlatforms Method (string, DateTime, DateTime, AnalyticsCriteria) 1166
GetReferringSites Method 1167
Analytics.GetReferringSites Method (DateTime, DateTime, AnalyticsCriteria) 1167
Analytics.GetReferringSites Method (string, DateTime, DateTime, AnalyticsCriteria) 1168
GetScreenColors Method 1168
Analytics.GetScreenColors Method (DateTime, DateTime, AnalyticsCriteria) 1168
Analytics.GetScreenColors Method (string, DateTime, DateTime, AnalyticsCriteria) 1169
GetScreenResolutions Method 1170
Analytics.GetScreenResolutions Method (DateTime, DateTime, AnalyticsCriteria) 1170
Analytics.GetScreenResolutions Method (string, DateTime, DateTime, AnalyticsCriteria) 1170
GetSearchEngines Method 1171
Analytics.GetSearchEngines Method (DateTime, DateTime, AnalyticsCriteria) 1171
Analytics.GetSearchEngines Method (string, DateTime, DateTime, AnalyticsCriteria) 1172
GetTimeOnSiteTrend Method 1172
Analytics.GetTimeOnSiteTrend Method (DateTime, DateTime, AnalyticsCriteria) 1173
Analytics.GetTimeOnSiteTrend Method (string, DateTime, DateTime, AnalyticsCriteria) 1173
GetTopContent Method 1174
Analytics.GetTopContent Method (DateTime, DateTime, AnalyticsCriteria) 1174
Analytics.GetTopContent Method (string, DateTime, DateTime, AnalyticsCriteria) 1175
GetTopExitPages Method 1175
Analytics.GetTopExitPages Method (DateTime, DateTime, AnalyticsCriteria) 1176
Analytics.GetTopExitPages Method (string, DateTime, DateTime, AnalyticsCriteria) 1176
GetTopLandingPages Method 1177
Analytics.GetTopLandingPages Method (DateTime, DateTime, AnalyticsCriteria) 1177
Analytics.GetTopLandingPages Method (string, DateTime, DateTime, AnalyticsCriteria) 1178
GetUserDefinedValue Method 1178
Analytics.GetUserDefinedValue Method (DateTime, DateTime, AnalyticsCriteria) 1178
Analytics.GetUserDefinedValue Method (string, DateTime, DateTime, AnalyticsCriteria) 1179
GetVisitsTrend Method 1180
Analytics.GetVisitsTrend Method (DateTime, DateTime, AnalyticsCriteria) 1180
Analytics.GetVisitsTrend Method (string, DateTime, DateTime, AnalyticsCriteria) 1180
CommunityGroup Class 1181

CommunityGroup Methods 1181
CommunityGroup.AddCommunityGroup Method 1182
AddUserToCommunityGroup Method 1183
CommunityGroup.AddUserToCommunityGroup Method (long, long) 1183
CommunityGroup.AddUserToCommunityGroup Method (long, long, bool) 1184
CommunityGroup.ApprovePendingGroupUser Method 1184
CommunityGroup.CancelJoinRequestForCommunityGroup Method 1185
CommunityGroup.DeleteCommunityGroupByID Method 1185
CommunityGroup.DeletePendingGroupUser Method 1186
CommunityGroup.GetAllCommunityGroups Method 1186
CommunityGroup.GetAllCommunityGroupsInDirectory Method 1186
CommunityGroup.GetAllUnassignedCommunityGroupUsers Method 1187
CommunityGroup.GetCommunityGroupBlogIds Method 1187
CommunityGroup.GetCommunityGroupByID Method 1188
CommunityGroup.GetCommunityGroupDiscussionBoardIds Method 1188
CommunityGroup.GetCommunityGroups Method 1189
CommunityGroup.GetCommunityGroupUserIDs Method 1189
GetCommunityGroupUsers Method 1190
CommunityGroup.GetCommunityGroupUsers Method (long) 1190
CommunityGroup.GetCommunityGroupUsers Method (long, PagingInfo) 1191
CommunityGroup.GetFolderIdForCommunityGroup Method 1191
CommunityGroup.GetGroupMemberStatus Method 1192
CommunityGroup.GetInvitedCommunityGroups Method 1192
GetMostPopularCommunityGroups Method 1193
CommunityGroup.GetMostPopularCommunityGroups Method (PagingInfo) 1193
CommunityGroup.GetMostPopularCommunityGroups Method (int) 1194
CommunityGroup.GetPendingCommunityGroups Method 1194
CommunityGroup.GetPendingCommunityGroupUsers Method 1195
CommunityGroup.InviteUsers Method 1195
CommunityGroup.IsCommunityGroupAdmin Method 1196
CommunityGroup.IsUserInCommunityGroup Method 1196
CommunityGroup.RemoveUserFromCommunityGroup Method 1197
CommunityGroup.UpdateCommunityGroup Method 1197
Content Class 1198
Content Methods 1198
AddContent Method 1199
Content.AddContent Method (string, string, string, string, string, string, long, object, object, string) 1200
Content.AddContent Method (string, string, string, string, string, string, long, object, object, string, long) 1201

Content.AddContent Method (string, string, string, string, string, string, long, object, object, string, long, long) 1202
Content.AddContent Method (string, string, string, string, string, string, string, string, string, long, object, object, string) 1202
Content.CheckOutContent Method 1203
Content.CopyContentToFolder Method 1203
Content.DeleteContentItem Method 1204
Content.GetAddViewLanguage Method 1204
Content.GetAllComments Method 1205
Content.GetAllTemplates Method 1205
Content.GetAllUnassignedItemApprovals Method 1206
GetChildContent Method 1206
Content.GetChildContent Method (long, bool, string) 1206
Content.GetChildContent Method (long, bool, string, PagingInfo) 1207
Content.GetContent Method 1208
Content.GetContentByHistory Method 1208
Content.GetContentForEditing Method 1209
Content.GetContentState Method 1209
Content.GetContentStatus Method 1210
Content.GetDomain Method 1210
Content.MoveContentToFolder Method 1210
Content.PublishContent Method 1211
Content.RenameContent Method 1211
Content.SaveContent Method 1212
Content.SubmitForDelete Method 1213
Content.UndoCheckout Method 1213
Content.UpdateContentMetaData Method 1213
ContentRating Class 1214
ContentRating Methods 1214
ContentRating.AddContentRating Method 1215
ContentRating.GetContentRatingResults Method 1215
ContentRating.PurgeContentRatings Method 1216
CustomFields Class 1216
CustomFields Methods 1217
CustomFields.AddNewEditableCustomFieldAssignments Method 1218
CustomFields.BoolToYesNo Method 1218
CustomFields.GetEditableCustomFieldAssignments Method 1219
CustomFields.GetEnhancedMetadataArea Method 1220
CustomFields.GetEnhancedMetadataScript Method 1220
CustomFields.GetFieldsByFolder Method 1220

CustomFields.GetSearchProperties Method 1221
CustomFields.ProcessCustomFields Method 1221
CustomFields.SearchAllAssets Method 1222
CustomFields.WriteFilteredMetadataForEdit Method 1222
CustomFields.WriteFilteredMetadataForView Method 1223
CustomFields.WriteLibrarySearchExtended Method 1223
CustomFields.WriteMetadataDefaultForEdit Method 1224
CustomFields.WriteMetadataTypeForView Method 1224
CustomFields.WritePublicSearch Method 1225
Favorites Class 1225
Favorites Methods 1226
AddContentFavorite Method 1226
Favorites.AddContentFavorite Method (long, long) 1226
Favorites.AddContentFavorite Method (long, long, long) 1227
Favorites.DeleteContentFavorite Method 1227
GetContentFavorites Method 1228
Favorites.GetContentFavorites Method (long, PagingInfo) 1228
Favorites.GetContentFavorites Method (long, long) 1229
Favorites.GetContentFavorites Method (long, long, PagingInfo) 1229
Favorites.GetFavoritesDirectory Method 1230
Favorites.IsContentFavorite Method 1230
Favorites.MoveContentFavorite Method 1231
Flag Class 1231
Flag Methods 1232
Flag.AddFlagObject Method 1232
Flag.AssignFlagToContent Method 1233
Flag.GetAllFlagEntries Method 1233
Flag.GetContentFlagData Method 1234
Flag.GetFlagDataById Method 1234
Flag.GetFlaggedRank Method 1235
Flag.PurgeFlagEntries Method 1235
Flag.RemoveObjectObject Method 1235
Flag.UpdateFlagObject Method 1236
Folder Class 1236
Folder Methods 1237
Folder.AddFolder Method 1237
Folder.CreateFolder Method 1238
Folder.DeleteFolder Method 1238
GetChildFolders Method 1238

Folder.GetChildFolders Method (long, bool, Ektron.Cms.Common.EkEnumeration.FolderOrderBy) 1239
Folder.GetChildFolders Method (long, bool, string) 1239
GetFolder Method 1240
Folder.GetFolder Method (long) 1240
Folder.GetFolder Method (long, bool, bool) 1241
GetFolderID Method 1241
Folder.GetFolderID Method (long) 1241
Folder.GetFolderID Method (string, long) 1242
Folder.GetPath Method 1242
Folder.RenameFolder Method 1243
UpdateFolder Method 1243
Folder.UpdateFolder Method (FolderData) 1243
Folder.UpdateFolder Method (FolderRequest) 1244
Font Class 1244
Font Methods 1245
Font.AddFont Method 1245
Font.DeleteFont Method 1245
Font.GetAllFonts Method 1246
Font.GetFont Method 1246
Font.GetFontConfigList Method 1247
Font.UpdateFont Method 1247
Form Class 1247
Form Methods 1248
Form.AssignFormTask Method 1248
Form.EmailFormFieldData Method 1249
GetAllForms Method 1249
Form.GetAllForms Method () 1250
Form.GetAllForms Method (long, string, int, ContentAPI.FormResultType) 1250
Form.GetForm Method 1251
GetFormFieldData Method 1251
Form.GetFormFieldData Method (long, long) 1252
Form.GetFormFieldData Method (long, string, string) 1252
Form.GetFormFieldList Method 1253
Form.GetFormFieldQuestions Method 1253
Form.GetFormTitle Method 1254
Form.SubmitFormFieldData Method 1254
Form.UpdateFormFieldData Method 1255
Friends Class 1255

Friends Methods 1256
Friends.AcceptPendingFriend Method 1257
AddFriend Method 1257
Friends.AddFriend Method (long, long) 1257
Friends.AddFriend Method (long, long, long) 1258
Friends.AddPendingFriend Method 1258
Friends.AddSelectedFriend Method 1259
Friends.DeleteFriend Method 1259
DeletePendingFriendRequest Method 1260
Friends.DeletePendingFriendRequest Method (long, long) 1260
Friends.DeletePendingFriendRequest Method (long, string) 1260
Friends.DeleteSelectedFriend Method 1261
GetFriends Method 1261
Friends.GetFriends Method (long) 1262
Friends.GetFriends Method (long, PagingInfo) 1262
Friends.GetFriends Method (long, PagingInfo, long) 1263
Friends.GetFriends Method (long, long) 1263
Friends.GetFriendsDirectory Method 1264
Friends.GetFriendStatus Method 1264
Friends.GetInvitedUsers Method 1265
Friends.GetPendingFriends Method 1266
Friends.GetSelectedFriends Method 1266
Friends.GetSentFriendInvitations Method 1267
Friends.InviteUsers Method 1267
Friends.IsFriend Method 1268
Friends.IsSelectedFriend Method 1268
Friends.MoveFriend Method 1269
Library Class 1269
Library Methods 1270
Library.AddLoadBalanceItem Method 1270
Library.DeleteLibraryItem Method 1271
Library.DeleteLibraryItemById Method 1271
Library.DeleteLoadBalanceItem Method 1272
Library.GetAllChildLibItems Method 1272
GetAllLoadBalancePaths Method 1273
Library.GetAllLoadBalancePaths Method (long, string) 1273
Library.GetAllLoadBalancePaths Method (string) 1274
GetLibraryItem Method 1274
Library.GetLibraryItem Method (long) 1274

Library.GetLibraryItem Method (long, long) 1275
Library.GetLibrarySettings Method 1275
Library.GetLibraryTypes Method 1276
Library.GetLoadBalancePath Method 1276
Library.InternalLibrarySearch Method 1277
Library.UpdateLibrary Method 1277
Library.UpdateLibrarySettings Method 1278
Library.UpdateLoadBalanceSettings Method 1278
Library.UpdateQlinkTemplates Method 1278
MessageBoard Class 1279
MessageBoard Methods 1279
MessageBoard.AddMessageBoardEntry Method 1280
ApproveMessage Method 1280
MessageBoard.ApproveMessage Method (long, long, EkEnumeration.MessageBoardObjectType) 1281
MessageBoard.ApproveMessage Method (long, long, EkEnumeration.MessageBoardObjectType, long) 1281
MessageBoard.ApproveMessage Method (long, long, EkEnumeration.MessageBoardObjectType, long, bool) 1282
DeleteMessageBoardEntries Method 1283
MessageBoard.DeleteMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType) 1283
MessageBoard.DeleteMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, long) 1284
MessageBoard.DeleteMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, long, bool) 1284
DeleteMessageBoardEntriesByUser Method 1285
MessageBoard.DeleteMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long) 1285
MessageBoard.DeleteMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, long) 1286
MessageBoard.DeleteMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, long, bool) 1287
DeleteMessageBoardEntry Method 1287
MessageBoard.DeleteMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType) 1288
MessageBoard.DeleteMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long) 1288
MessageBoard.DeleteMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long, bool) 1289
GetAllMessageBoardEntries Method 1289
MessageBoard.GetAllMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, PagingInfo) 1290

MessageBoard.GetAllMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, PagingInfo, long) 1291
MessageBoard.GetAllMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, PagingInfo, long, bool) 1291
GetAllMessageBoardEntriesByUser Method 1292
MessageBoard.GetAllMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo) 1293
MessageBoard.GetAllMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo, long) 1293
MessageBoard.GetAllMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo, long, bool) 1294
GetMessageBoardEntry Method 1294
MessageBoard.GetMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType) 1295
MessageBoard.GetMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long) 1296
MessageBoard.GetMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long, bool) 1296
MessageBoard.IsUserMessageBoardAdmin Method 1297
UpdateMessageText Method 1297
MessageBoard.UpdateMessageText Method (long, long, EkEnumeration.MessageBoardObjectType, string) 1298
MessageBoard.UpdateMessageText Method (long, long, EkEnumeration.MessageBoardObjectType, string, long) 1298
MessageBoard.UpdateMessageText Method (long, long, EkEnumeration.MessageBoardObjectType, string, long, bool) 1299
Metadata Class 1300
Metadata Methods 1300
Metadata.AddMetaDataType Method 1301
Metadata.DeleteMetadataType Method 1301
GetContentMetadataList Method 1301
Metadata.GetContentMetadataList Method (long) 1302
Metadata.GetContentMetadataList Method (long, bool) 1302
Metadata.GetMetadataType Method 1303
Metadata.GetMetaDataTypes Method 1303
Metadata.UpdateMetadataType Method 1304
Permissions Class 1304
Permissions Methods 1305
Permissions.AddItemPermission Method 1305
Permissions.DeleteItemPermission Method 1306
Permissions.DisableContentInheritance Method 1306
Permissions.DisableFolderInheritance Method 1307
Permissions.EnableContentInheritance Method 1307

Permissions.EnableFolderInheritance Method 1308
GetUserPermissions Method 1308
Permissions.GetUserPermissions Method (long, string, long, string) 1309
Permissions.GetUserPermissions Method (long, string, long, string, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) 1309
Permissions.LoadPermissions Method 1310
Permissions.UpdateItemPermission Method 1310
SearchManager Class 1311
SearchManager Methods 1311
ExecuteSearch Method 1312
SearchManager.ExecuteSearch Method (ContentSearchCondition) 1312
SearchManager.ExecuteSearch Method (UserSearchCondition) 1312
Search Method 1313
SearchManager.Search Method (MapRequestData, int, int, int) 1313
SearchManager.Search Method (SearchRequestData, int) 1314
SearchManager.Search Method (string, int) 1314
SearchManager.SearchForCategories Method 1315
SearchManager.SearchTable Method 1315
Site Class 1316
Site Methods 1316
Site.BreakTaskPermissionInherit Method 1317
Site.GetAllActiveLanguages Method 1317
GetAllLanguages Method 1318
Site.GetAllLanguages Method () 1318
Site.GetAllLanguages Method (LanguageDataSet.SortBy, LanguageDataSet.SortDirection) 1318
Site.GetClientDayNames Method 1319
Site.GetClientMonthNames Method 1319
Site.GetCookieValue Method 1320
Site.GetLanguage Method 1320
Site.GetLanguages Method 1321
Site.GetSiteVariables Method 1321
Site.GetTaskPermission Method 1322
Site.IsTaskPermissionInherited Method 1322
Site.SetCookieValue Method 1323
Site.SetTaskPermission Method 1323
Site.UpdateSiteEnabledLanguages Method 1324
SiteMap Class 1324
SiteMap Methods 1325
SiteMap.GetFolderBreadcrumbPath Method 1325

SiteMap.GetWebsiteMap Method 1325
SiteMap.RenderFolderBreadcrumbHtml Method 1326
SiteMap.RenderWebsiteMap Method 1327
Tags Class 1327
Tags Methods 1328
Tags.AddTag Method 1329
AddTagToCommunityGroup Method 1329
Tags.AddTagToCommunityGroup Method (string, long) 1330
Tags.AddTagToCommunityGroup Method (string, long, int) 1331
Tags.AddTagToCommunityGroup Method (string, long, long, int) 1331
AddTagToContent Method 1332
Tags.AddTagToContent Method (string, long) 1332
Tags.AddTagToContent Method (string, long, long, int) 1333
AddTagToObject Method 1333
Tags.AddTagToObject Method (string, long, EkEnumeration.CMSObjectTypes) 1334
Tags.AddTagToObject Method (string, long, EkEnumeration.CMSObjectTypes, long, int) 1334
AddTagToUser Method 1335
Tags.AddTagToUser Method (long, long) 1336
Tags.AddTagToUser Method (string, long) 1336
Tags.AddTagToUser Method (string, long, int) 1337
Tags.DeleteTagById Method 1337
Tags.DeleteTagOnObject Method 1338
Tags.DeleteTagsByUser Method 1338
Tags.DeleteTagsOnCommunityGroup Method 1339
Tags.DeleteTagsOnContent Method 1339
Tags.DeleteTagsOnObject Method 1340
Tags.DeleteTagsOnUser Method 1340
Tags.EditTag Method 1341
Tags.GetAllTagRelatedContent Method 1341
Tags.GetAllTagRelatedUsers Method 1342
GetAllTags Method 1342
Tags.GetAllTags Method (PagingInfo) 1343
Tags.GetAllTags Method (TagRequestData, int) 1343
GetDefaultTags Method 1344
Tags.GetDefaultTags Method (EkEnumeration.CMSObjectTypes) 1344
Tags.GetDefaultTags Method (EkEnumeration.CMSObjectTypes, int) 1345
Tags.GetTagByID Method 1345
Tags.GetTagByName Method 1346
Tags.GetTagCloud Method 1346

Tags.GetTagsForContent Method 1347
GetTagsForObject Method 1347
Tags.GetTagsForObject Method (long, EkEnumeration.CMSObjectTypes) 1347
Tags.GetTagsForObject Method (long, EkEnumeration.CMSObjectTypes, int) 1348
GetTagsForUser Method 1348
Tags.GetTagsForUser Method (long) 1349
Tags.GetTagsForUser Method (long, int) 1349
Tags.RemoveTagAsDefault Method 1350
Tags.SaveTagAsDefault Method 1350
Taxonomy Class 1351
Taxonomy Methods 1351
AddTaxonomyItem Method 1352
Taxonomy.AddTaxonomyItem Method (TaxonomyContentRequest) 1352
Taxonomy.AddTaxonomyItem Method (TaxonomyRequest) 1353
Taxonomy.AddTaxonomySyncFolder Method 1353
Taxonomy.CreateTaxonomy Method 1353
Taxonomy.DeleteTaxonomy Method 1354
Taxonomy.GetTaxonomyIdByPath Method 1354
Taxonomy.LoadTaxonomy Method 1355
Taxonomy.ReadAllAssignedCategory Method 1355
Taxonomy.ReadTaxonomy Method 1356
Taxonomy.RemoveTaxonomyFolder Method 1356
Taxonomy.RemoveTaxonomyItem Method 1357
Taxonomy.ReOrderTaxonomyItems Method 1357
Taxonomy.UpdateTaxonomy Method 1357
ThreadedDiscussion Class 1358
ThreadedDiscussion Methods 1358
ThreadedDiscussion.AddBoard Method 1359
ThreadedDiscussion.AddForum Method 1360
ThreadedDiscussion.AddReply Method 1360
AddTopic Method 1361
ThreadedDiscussion.AddTopic Method (long, string, string) 1361
ThreadedDiscussion.AddTopic Method (long, string, string, int) 1362
ThreadedDiscussion.DeleteBoard Method 1362
ThreadedDiscussion.DeleteForum Method 1363
ThreadedDiscussion.DeleteTopic Method 1363
ThreadedDiscussion.GetBoardCategories Method 1364
ThreadedDiscussion.GetBoardForums Method 1364
GetDiscussionBoard Method 1364

ThreadedDiscussion.GetDiscussionBoard Method (long) 1365
ThreadedDiscussion.GetDiscussionBoard Method (string) 1365
ThreadedDiscussion.GetDiscussionBoardString Method 1366
ThreadedDiscussion.GetForum Method 1366
ThreadedDiscussion.GetRepliesForTopics Method 1367
ThreadedDiscussion.GetTopic Method 1368
ThreadedDiscussion.SearchReplies Method 1368
User Class 1369
User Methods 1369
ActivateUserAccount Method 1371
User.ActivateUserAccount Method (long) 1372
User.ActivateUserAccount Method (string, string) 1372
User.AddMembershipUser Method 1373
User.AddNewUser Method 1373
User.AddRolePermission Method 1374
User.AddUser Method 1374
User.AddUserGroup Method 1375
User.AddUserToGroup Method 1375
User.AutoLogInUser Method 1376
CreateBlankUserObject Method 1376
User.CreateBlankUserObject Method () 1376
User.CreateBlankUserObject Method (UserData) 1377
User.DeleteUserByID Method 1377
User.DeleteUserByIds Method 1378
User.DeleteUserFromGroup Method 1378
User.DeleteUserGroup Method 1379
User.EditUserCustomProperties Method 1379
User.GetActiveUser Method 1380
User.GetADMapping Method 1380
User.GetAllCmsUsers Method 1381
User.GetAllCustomProperties Method 1381
User.GetAllUserByCustomProperty Method 1382
GetAllUserGroups Method 1382
User.GetAllUserGroups Method (EkEnumeration.UserTypes, string) 1383
User.GetAllUserGroups Method (string) 1383
GetAllUsers Method 1384
User.GetAllUsers Method (UserRequestData) 1384
User.GetAllUsers Method (long, long) 1385
User.GetGroupsUserIsIn Method 1385

User.GetMembershipUser Method 1386
User.GetUser Method 1386
User.GetUserbyUsername Method 1387
User.GetUserCustomProperties Method 1387
User.GetUserEmailInfo Method 1388
User.GetUserForFolderAdmin Method 1388
User.GetUserGroup Method 1389
User.GetUserGroupForFolderAdmin Method 1389
User.GetUserPermissions Method 1390
User.GetUserPreference Method 1390
GetUsers Method 1391
User.GetUsers Method (long, string) 1391
User.GetUsers Method (string, string) 1392
User.GetUsersForTask Method 1393
User.GetUsersNotInGroup Method 1393
User.IsAdmin Method 1394
IsARoleMember Method 1394
User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) 1396
User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, long, bool) 1396
User.IsARoleMember Method (long) 1397
User.IsARoleMember Method (long, long, bool) 1397
User.IsARoleMemberForContent Method 1398
User.IsARoleMemberForFolder Method 1398
User.LogInUser Method 1399
ResetMembershipUserPassword Method 1400
User.ResetMembershipUserPassword Method (string) 1400
User.ResetMembershipUserPassword Method (string, string) 1400
User.UnsubscribeUser Method 1401
User.UpdateCustomPropertiesItemOrder Method 1401
User.UpdateMapCoordinate Method 1402
UpdateUser Method 1402
User.UpdateUser Method (Collection) 1403
User.UpdateUser Method (UserData) 1403
User.UpDateUserGroup Method 1404
User.UpdateUserPreferences Method 1404
UserObject Method 1405
User.UserObject Method (long) 1405
User.UserObject Method (string) 1405

Providers API **1406**
Ektron.Cms.Commerce.Inventory Namespace 1407
Ektron.Cms.Commerce.Inventory.Provider Namespace 1407
InventoryProvider Class 1407
InventoryProvider Methods 1407
InventoryProvider.DecreaseStockLevel Method 1408
InventoryProvider.GetInventory Method 1409
InventoryProvider.GetReorderLevel Method 1409
InventoryProvider.GetUnitsInStock Method 1410
InventoryProvider.GetUnitsOnOrder Method 1410
InventoryProvider.IncreaseStockLevel Method 1410
InventoryProvider.SaveInventory Method 1411
PaymentGatewayProvider Class 1412
PaymentGatewayProvider Methods 1412
PaymentGatewayProvider.Authorize Method 1412
PaymentGatewayProvider.AuthorizeAndCapture Method 1413
PaymentGatewayProvider.AuthorizeRecurringPayment Method 1413
PaymentGatewayProvider.CancelRecurringPayment Method 1413
PaymentGatewayProvider.CapturePreauthorization Method 1414
PaymentGatewayProvider.RecurringPaymentFailed Method 1414
PaymentGatewayProvider.RecurringPaymentProcessed Method 1414
PaymentGatewayProvider.VoidPreAuthorization Method 1414
PaymentGatewayProvider Properties 1415
PaymentGatewayProvider.Amount Property 1415
PaymentGatewayProvider.Authorization Property 1416
PaymentGatewayProvider.Config Property 1416
PaymentGatewayProvider.CurrencyCode Property 1416
PaymentGatewayProvider.CurrencyId Property 1416
PaymentGatewayProvider.GatewayData Property 1416
PaymentGatewayProvider.InTestMode Property 1416
PaymentGatewayProvider.IsSubmissionSuccess Property 1417
PaymentGatewayProvider.OrderId Property 1417
PaymentGatewayProvider.PaymentMethod Property 1417
PaymentGatewayProvider.SubmissionError Property 1417
PaymentGatewayProvider.SupportsCheckPayments Property 1417
PaymentGatewayProvider.SupportsCreditCardPayments Property 1418
PaymentGatewayProvider.SupportsRecurringPayments Property 1418
Ektron.Cms.Commerce.Shipment Namespace 1418
Ektron.Cms.Commerce.Shipment.Provider Namespace 1418

ShipmentProvider Class 1419
ShipmentProvider Methods 1419
ShipmentProvider.GetRates Method 1419
ShipmentProvider.GetServiceTypes Method 1420
ShipmentProvider.GetTrackingUrl Method 1420
ShipmentProvider Properties 1420
ShipmentProvider.AccountNumber Property 1421
ShipmentProvider.IsTrackingSupported Property 1421
ShipmentProvider.Key Property 1421
ShipmentProvider.MeterNumber Property 1421
ShipmentProvider.Password Property 1421
ShipmentProvider.ServiceUrl Property 1422
ShipmentProvider.TransactionId Property 1422

Data Classes 1423

Ektron.Cms.Commerce Namespace 1423
Classes 1426
AddressData Class 1428
AddressData Properties 1429
AddressData.AddressLine1 Property 1429
AddressData.AddressLine2 Property 1429
AddressData.City Property 1429
AddressData.Company Property 1430
AddressData.Country Property 1430
AddressData.Id Property 1430
AddressData.Name Property 1430
AddressData.Phone Property 1430
AddressData.PostalCode Property 1430
AddressData.Region Property 1431
AdjustedBasketItem Class 1431
AdjustedBasketItem Constructor 1431
AdjustedBasketItem.AdjustedBasketItem Constructor () 1431
AdjustedBasketItem.AdjustedBasketItem Constructor (BasketItem) 1432
AdjustedBasketItem Properties 1432
AdjustedBasketItem.AdjustedTotal Property 1432
AdjustedBasketItem.ApplicableTaxRate Property 1432
AdjustedBasketItem.AppliedCoupons Property 1433
AdjustedBasketItem.HasFreeShipping Property 1433

AdjustedBasketItem.TaxesApplied Property 1433
AdjustedBasketItem.TotalCouponDiscount Property 1433
AppliedCouponData Class 1433
AppliedCouponData Constructor 1434
AppliedCouponData.AppliedCouponData Constructor () 1434
AppliedCouponData.AppliedCouponData Constructor (CouponData, decimal) 1434
AppliedCouponData Properties 1434
AppliedCouponData.Coupon Property 1434
AppliedCouponData.CouponDiscount Property 1435
AppliedCouponDataCollection Class 1435
AppliedCouponDataCollection Fields 1435
AppliedCouponDataCollection.ItemAdded Field 1435
AppliedCouponDataCollection Methods 1435
AppliedCouponDataCollection.FindByCouponId Method 1436
AppliedCouponEventArgs Class 1436
AppliedCouponEventArgs Constructor 1436
AppliedCouponEventArgs.AppliedCouponEventArgs Constructor () 1436
AppliedCouponEventArgs.AppliedCouponEventArgs Constructor (AppliedCouponData) 1437
AppliedCouponEventArgs Properties 1437
AppliedCouponEventArgs.AppliedCoupon Property 1437
Basket Class 1437
Basket Methods 1438
AddProduct Method 1438
Basket.AddProduct Method (long) 1438
Basket.AddProduct Method (long, KitConfigData) 1439
Basket.AddProduct Method (long, int) 1439
Basket.AddProductVariant Method 1439
Basket.Decrement Method 1439
Basket.GetItemIndex Method 1440
Basket.GetKitConfiguration Method 1440
Basket.GetKitConfigurationAsAdmin Method 1440
Basket.IncrementItemQuantity Method 1440
Basket.MakeDefaultBasket Method 1440
Basket.RemoveItem Method 1441
Basket.SaveBasket Method 1441
Basket.UpdateItemQuantity Method 1441
Basket.UpdateKitConfiguration Method 1441
Basket Properties 1442
Basket.RequestInformation Property 1442

BasketCalculatorData Class 1442
BasketCalculatorData Constructor 1442
BasketCalculatorData.BasketCalculatorData Constructor () 1443
BasketCalculatorData.BasketCalculatorData Constructor (BasketData) 1443
BasketCalculatorData Properties 1443
BasketCalculatorData.AdjustedItems Property 1444
BasketCalculatorData.AllBasketCoupons Property 1444
BasketCalculatorData.Basket Property 1444
BasketCalculatorData.BasketCoupons Property 1444
BasketCalculatorData.BasketTotal Property 1444
BasketCalculatorData.HasFreeShipping Property 1444
BasketCalculatorData.ShipFrom Property 1445
BasketCalculatorData.ShippingRates Property 1445
BasketCalculatorData.ShipTo Property 1445
BasketCalculatorData.TaxClassSubtotals Property 1445
BasketCalculatorData.TotalCouponDiscount Property 1445
BasketCalculatorData.TotalShippingCost Property 1446
BasketCalculatorData.TotalTaxes Property 1446
BasketCalculatorData.ValidationResults Property 1446
BasketData Class 1446
BasketData Properties 1446
BasketData.DateCreated Property 1447
BasketData.DateModified Property 1447
BasketData.Id Property 1447
BasketData.IsDefault Property 1447
BasketData.Items Property 1447
BasketData.Name Property 1448
BasketData.ShippingAddressId Property 1448
BasketData.ShippingMethodId Property 1448
BasketData.ShopperId Property 1448
BasketData.Subtotal Property 1448
BasketData.UserId Property 1449
BasketItem Class 1449
BasketItem Constructor 1449
BasketItem.BasketItem Constructor () 1449
BasketItem.BasketItem Constructor (BasketItem) 1449
BasketItem Properties 1450
BasketItem.AdjustedTotal Property 1450
BasketItem.Configuration Property 1450

BasketItem.Dimensions Property 1451
BasketItem.Id Property 1451
BasketItem.IsInventoryManagementDisabled Property 1451
BasketItem.IsTangible Property 1451
BasketItem.ListPrice Property 1451
BasketItem.Pricing Property 1452
BasketItem.ProductId Property 1452
BasketItem.ProductLanguage Property 1452
BasketItem.ProductSku Property 1452
BasketItem.ProductTitle Property 1452
BasketItem.ProductType Property 1453
BasketItem.Quantity Property 1453
BasketItem.Quicklink Property 1453
BasketItem.SalePrice Property 1453
BasketItem.TaxClassId Property 1453
BasketItem.Variant Property 1453
BasketItem.Weight Property 1454
BasketItemCollection Class 1454
BasketItemCollection.BasketItemCollection Constructor 1454
BasketItemCollection Methods 1454
BasketItemCollection.AddRange Method 1455
BasketItemCollection.Find Method 1455
BasketItemCollection.FindAll Method 1455
BasketItemCollection.FindIndex Method 1455
BasketVariantData Class 1455
BasketVariantData.BasketVariantData Constructor 1456
BasketVariantData Properties 1456
BasketVariantData.Dimensions Property 1456
BasketVariantData.Id Property 1456
BasketVariantData.Language Property 1457
BasketVariantData.ListPrice Property 1457
BasketVariantData.Pricing Property 1457
BasketVariantData.SalePrice Property 1457
BasketVariantData.Sku Property 1457
BasketVariantData.TaxClassId Property 1458
BasketVariantData.Title Property 1458
BasketVariantData.Weight Property 1458
BundleData Class 1458
BundleData.BundleData Constructor 1458

BundleData Properties 1459
BundleData.BundledItems Property 1459
CatalogData Class 1459
CatalogData Properties 1459
CatalogData.Id Property 1460
CatalogData.Name Property 1460
ContractData Class 1460
ContractData Constructor 1460
ContractData.ContractData Constructor () 1461
ContractData.ContractData Constructor (string) 1461
ContractData Properties 1461
ContractData.Id Property 1461
ContractData.Name Property 1461
CountryData Class 1461
CountryData Constructor 1462
CountryData.CountryData Constructor () 1462
CountryData.CountryData Constructor (int, string, string, string, bool) 1462
CountryData Methods 1463
CountryData.ToString Method 1463
CountryData.Validate Method 1463
CountryData Properties 1463
CountryData.Enabled Property 1463
CountryData.Id Property 1464
CountryData.LongIsoCode Property 1464
CountryData.Name Property 1464
CountryData.ShortIsoCode Property 1464
CountryTaxRateData Class 1464
CountryTaxRateData Constructor 1465
CountryTaxRateData.CountryTaxRateData Constructor () 1465
CountryTaxRateData.CountryTaxRateData Constructor (long, long, decimal) 1465
CountryTaxRateData.CountryTaxRateData Constructor (long, long, decimal, long) 1465
CountryTaxRateData Properties 1466
CountryTaxRateData.CountryData Property 1466
CouponData Class 1466
CouponData.CouponData Constructor 1466
CouponData Properties 1466
CouponData.Code Property 1467
CouponData.CouponType Property 1467
CouponData.CurrencyId Property 1467

CouponData.Description Property 1468
CouponData.DiscountType Property 1468
CouponData.DiscountValue Property 1468
CouponData.ExpirationDate Property 1468
CouponData.Id Property 1468
CouponData.IsActive Property 1468
CouponData.IsCombinable Property 1469
CouponData.IsRedeemable Property 1469
CouponData.MaximumAmount Property 1469
CouponData.MaximumUses Property 1469
CouponData.MinimumAmount Property 1469
CouponData.OnePerCustomer Property 1469
CouponData.StartDate Property 1470
CouponEntryData Class 1470
CouponEntryData Constructor 1470
CouponEntryData.CouponEntryData Constructor () 1470
CouponEntryData.CouponEntryData Constructor (long, EkEnumeration.CMSObjectTypes, long, string, bool) 1471
CouponEntryData Properties 1471
CouponEntryData.CouponId Property 1471
CouponEntryData.DataState Property 1471
CouponEntryData.IsIncluded Property 1472
CouponEntryData.ObjectId Property 1472
CouponEntryData.ObjectType Property 1472
CouponEntryData.Path Property 1472
CouponEntryData.Title Property 1472
CurrencyData Class 1472
CurrencyData Properties 1473
CurrencyData.AlphaIsoCode Property 1473
CurrencyData.CultureCode Property 1473
CurrencyData.CurrencySymbol Property 1473
CurrencyData.Enabled Property 1474
CurrencyData.Id Property 1474
CurrencyData.ISOCurrencySymbol Property 1474
CurrencyData.Name Property 1474
CurrencyPricingData Class 1474
CurrencyPricingData Constructor 1475
CurrencyPricingData.CurrencyPricingData Constructor () 1475
CurrencyPricingData.CurrencyPricingData Constructor (long, decimal, EkEnumeration.PricingType) 1475

CurrencyPricingData.CurrencyPricingData Constructor (long, decimal, decimal) 1475
CurrencyPricingData Methods 1476
CurrencyPricingData.Clone Method 1476
CurrencyPricingData.GetSalePrice Method 1476
CurrencyPricingData.Validate Method 1476
CurrencyPricingData Properties 1477
CurrencyPricingData.AlphaIsoCode Property 1477
CurrencyPricingData.CurrencyId Property 1477
CurrencyPricingData.ListPrice Property 1477
CurrencyPricingData.NumericIsoCode Property 1477
CurrencyPricingData.PricingType Property 1478
CurrencyPricingData.SalePrice Property 1478
CurrencyPricingData.TierPrices Property 1478
CustomerData Class 1478
CustomerData Properties 1478
CustomerData.AverageOrderValue Property 1479
CustomerData.BillingAddressId Property 1479
CustomerData.DisplayName Property 1479
CustomerData.EmailAddress Property 1479
CustomerData.FirstName Property 1480
CustomerData.Id Property 1480
CustomerData.IsDeleted Property 1480
CustomerData.IsMembershipUser Property 1480
CustomerData.LastName Property 1480
CustomerData.Password Property 1480
CustomerData.ShippingAddressId Property 1481
CustomerData.TotalOrders Property 1481
CustomerData.TotalOrderValue Property 1481
CustomerData.UserId Property 1481
CustomerData.UserName Property 1481
Dimensions Class 1482
Dimensions Constructor 1482
Dimensions.Dimensions Constructor () 1482
Dimensions.Dimensions Constructor (LinearUnit, float, float, float) 1482
Dimensions Methods 1483
Dimensions.Clone Method 1483
Dimensions.CompareTo Method 1483
Dimensions.ConvertUnits Method 1483
Dimensions Properties 1484

Dimensions.CubicUnits Property 1484

Dimensions.Height Property 1484

Dimensions.Length Property 1484

Dimensions.Units Property 1484

Dimensions.Width Property 1484

EntryAttributeData Class 1485

EntryAttributeData Constructor 1485

EntryAttributeData.EntryAttributeData Constructor () 1485

EntryAttributeData.EntryAttributeData Constructor (string, int, EkEnumeration.ProductTypeAttributeDataType, object, EkEnumeration.ProductTypeAttributeStatus) 1486

EntryAttributeData.EntryAttributeData Constructor (string, long, object, int, EkEnumeration.ProductTypeAttributeDataType, string, EkEnumeration.ProductTypeAttributeStatus) 1486

EntryAttributeData Properties 1486

EntryAttributeData.AttributeTypeId Property 1487

EntryAttributeData.CurrentValue Property 1487

EntryAttributeData.VersionId Property 1487

EntryData Class 1487

EntryData Constructor 1487

EntryData.EntryData Constructor () 1488

EntryData.EntryData Constructor (EkEnumeration.CatalogEntryType) 1488

EntryData.EntryData Constructor (int, EkEnumeration.CatalogEntryType) 1488

EntryData Properties 1488

EntryData.Attributes Property 1490

EntryData.Comment Property 1490

EntryData.ContentStatus Property 1490

EntryData.CurrencyId Property 1490

EntryData.DateModified Property 1490

EntryData.Dimensions Property 1490

EntryData.DisableInventoryManagement Property 1491

EntryData.EndDate Property 1491

EntryData.EndDateAction Property 1491

EntryData.EntryType Property 1491

EntryData.FolderId Property 1492

EntryData.GoLive Property 1492

EntryData.Html Property 1492

EntryData.Id Property 1492

EntryData.Image Property 1493

EntryData.ImageThumbnail Property 1493

EntryData.IsArchived Property 1493
EntryData.IsBuyable Property 1493
EntryData.IsMarkedForDeletion Property 1493
EntryData.IsPublished Property 1494
EntryData.IsTangible Property 1494
EntryData.LanguageId Property 1494
EntryData.LastEditorFirstName Property 1494
EntryData.LastEditorId Property 1494
EntryData.LastEditorLastName Property 1495
EntryData.ListPrice Property 1495
EntryData.Media Property 1495
EntryData.Metadata Property 1495
EntryData.NumberSold Property 1495
EntryData.Pricing Property 1496
EntryData.ProductType Property 1496
EntryData.QuantityMultiple Property 1496
EntryData.Quicklink Property 1496
EntryData.SalePrice Property 1497
EntryData.Sku Property 1497
EntryData.Status Property 1497
EntryData.StatusLanguage Property 1497
EntryData.Summary Property 1497
EntryData.TaxClassId Property 1498
EntryData.TemplateId Property 1498
EntryData.Title Property 1498
EntryData.Url Property 1498
EntryData.VersionId Property 1498
EntryData.Weight Property 1499
EntryGroupData Class 1499
EntryGroupData Constructor 1499
EntryGroupData.EntryGroupData Constructor () 1499
EntryGroupData.EntryGroupData Constructor (long, string, string, long, EkEnumeration.EntryGroupType) 1500
EntryGroupData Properties 1500
EntryGroupData.Description Property 1500
EntryGroupData.DisplayOrder Property 1500
EntryGroupData.EntryId Property 1501
EntryGroupData.GroupType Property 1501
EntryGroupData.Id Property 1501

EntryGroupData.Name Property 1501
EntryGroupItemData Class 1501
EntryGroupItemData Constructor 1502
EntryGroupItemData.EntryGroupItemData Constructor () 1502
EntryGroupItemData.EntryGroupItemData Constructor (long, long, long) 1502
EntryGroupItemData Properties 1502
EntryGroupItemData.DisplayOrder Property 1502
EntryGroupItemData.EntryId Property 1503
EntryGroupItemData.GroupId Property 1503
EntryGroupItemData.Id Property 1503
ExchangeRateData Class 1503
ExchangeRateData.ExchangeRateData Constructor 1504
ExchangeRateData Methods 1504
ExchangeRateData.ConvertPrice Method 1504
ExchangeRateData.Validate Method 1504
ExchangeRateData Properties 1504
ExchangeRateData.BaseCurrencyId Property 1505
ExchangeRateData.EffectiveDate Property 1505
ExchangeRateData.ExchangeCurrencyId Property 1505
ExchangeRateData.Rate Property 1505
ImageMediaData Class 1506
ImageMediaData Properties 1506
ImageMediaData.Thumbnails Property 1506
InventoryData Class 1506
InventoryData Properties 1507
InventoryData.EntryId Property 1507
InventoryData.ReorderLevel Property 1507
InventoryData.UnitsInStock Property 1507
InventoryData.UnitsOnOrder Property 1507
KitConfigData Class 1508
KitConfigData.KitConfigData Constructor 1508
KitConfigData Methods 1508
KitConfigData.GetGroupIndex Method 1508
KitConfigData Properties 1508
KitConfigData.BasketItemId Property 1509
KitConfigData.Groups Property 1509
KitConfigData.Id Property 1509
KitConfigData.PriceModifier Property 1509
KitData Class 1510

KitData.KitData Constructor 1510
KitData Properties 1510
KitData.OptionGroups Property 1510
MediaData Class 1510
MediaData Properties 1511
MediaData.Alt Property 1511
MediaData.FileName Property 1511
MediaData.FilePath Property 1512
MediaData.Height Property 1512
MediaData.Id Property 1512
MediaData.MimeType Property 1512
MediaData.Sequence Property 1512
MediaData.Title Property 1513
MediaData.Width Property 1513
MediaGalleryData Class 1513
MediaGalleryData Properties 1513
MediaGalleryData.Images Property 1514
MediaGalleryData.ObjectId Property 1514
OptionGroupData Class 1514
OptionGroupData Constructor 1514
OptionGroupData.OptionGroupData Constructor () 1515
OptionGroupData.OptionGroupData Constructor (long, string, string, string, long) 1515
OptionGroupData.OptionGroupData Constructor (string, string) 1515
OptionGroupData Properties 1515
OptionGroupData.Image Property 1516
OptionGroupData.Options Property 1516
OptionGroupItemCollection Class 1516
OptionGroupItemCollection.OptionGroupItemCollection Constructor 1516
OptionGroupItemCollection Methods 1516
OptionGroupItemCollection.AddRange Method 1517
OptionGroupItemCollection.Find Method 1517
OptionGroupItemCollection.FindAll Method 1517
OptionGroupItemCollection.FindIndex Method 1517
OptionGroupItemData Class 1518
OptionGroupItemData Constructor 1518
OptionGroupItemData.OptionGroupItemData Constructor () 1518
OptionGroupItemData.OptionGroupItemData Constructor (long, long, string, decimal) 1518
OptionGroupItemData.OptionGroupItemData Constructor (string, decimal) 1519
OptionGroupItemData Properties 1519

OptionGroupItemData.CurrencyId Property 1519
OptionGroupItemData.ExtraText Property 1519
OptionGroupItemData.Name Property 1520
OptionGroupItemData.PriceModification Property 1520
OptionGroupItemData.PriceModificationType Property 1520
OrderData Class 1520
OrderData Properties 1520
OrderData.BillingAddressId Property 1521
OrderData.ClientIP Property 1521
OrderData.CouponTotal Property 1521
OrderData.CouponUsed Property 1522
OrderData.Currency Property 1522
OrderData.Customer Property 1522
OrderData.CustomerType Property 1522
OrderData.DateCompleted Property 1522
OrderData.DateCreated Property 1523
OrderData.DateRequired Property 1523
OrderData.HandlingTotal Property 1523
OrderData.HttpHost Property 1523
OrderData.Id Property 1523
OrderData.OrderTotal Property 1524
OrderData.Parts Property 1524
OrderData.ShippingTotal Property 1524
OrderData.SpecialInstructions Property 1524
OrderData.StageId Property 1524
OrderData.StageName Property 1525
OrderData.Status Property 1525
OrderData.Subtotal Property 1525
OrderData.TaxTotal Property 1525
OrderData.VisitorId Property 1525
OrderLineData Class 1526
OrderLineData Properties 1526
OrderLineData.Configuration Property 1526
OrderLineData.ItemType Property 1526
OrderLineData.LineItemId Property 1527
OrderLineData.OrderPartId Property 1527
OrderLineData.PriceEach Property 1527
OrderLineData.PriceSubTotal Property 1527
OrderLineData.PriceTotal Property 1527

OrderLineData.ProductId Property 1528
OrderLineData.ProductLanguage Property 1528
OrderLineData.Quantity Property 1528
OrderPartData Class 1528
OrderPartData Properties 1528
OrderPartData.DateShipped Property 1529
OrderPartData.HandlingTotal Property 1529
OrderPartData.Lines Property 1529
OrderPartData.OrderId Property 1530
OrderPartData.PartId Property 1530
OrderPartData.PartTotal Property 1530
OrderPartData.ShippingAddressId Property 1530
OrderPartData.ShippingTotal Property 1530
OrderPartData.Status Property 1531
OrderPartData.Subtotal Property 1531
OrderPartData.SupplierId Property 1531
OrderPartData.TaxTotal Property 1531
OrderPartData.TrackingNumber Property 1531
PackageData Class 1532
PackageData Constructor 1532
PackageData.PackageData Constructor () 1532
PackageData.PackageData Constructor (string, Dimensions, Weight) 1532
PackageData.PackageData Constructor (string, Dimensions, Weight, long) 1533
PackageData Properties 1533
PackageData.Dimensions Property 1533
PackageData.Id Property 1533
PackageData.MaxWeight Property 1533
PackageData.Name Property 1534
PaymentAuthorization Class 1534
PaymentAuthorization Properties 1534
PaymentAuthorization.AuthorizedOn Property 1534
PaymentAuthorization.CapturedOn Property 1535
PaymentAuthorization.CreditedOn Property 1535
PaymentAuthorization.Gateway Property 1535
PaymentAuthorization.Message Property 1535
PaymentAuthorization.RecurringTransactionId Property 1535
PaymentAuthorization.TransactionId Property 1535
PaymentAuthorization.VoidedOn Property 1536
PaymentMethod Class 1536

PaymentMethod Methods 1536
PaymentMethod.GetPayment Method 1536
PaymentMethod Properties 1537
PaymentMethod.BillingAddress Property 1537
PaymentMethod.BillingFirstName Property 1537
PaymentMethod.BillingLastName Property 1537
PipelineStageData Class 1537
PipelineStageData Properties 1538
PipelineStageData.Name Property 1538
PipelineStageData.Sequence Property 1538
PipelineStageData.StatusId Property 1538
PostalCodeTaxRateData Class 1539
PostalCodeTaxRateData Constructor 1539
PostalCodeTaxRateData.PostalCodeTaxRateData Constructor () 1539
PostalCodeTaxRateData.PostalCodeTaxRateData Constructor (string, long, long, decimal) 1539
PostalCodeTaxRateData.PostalCodeTaxRateData Constructor (string, long, long, decimal, long) 1540
PostalCodeTaxRateData Properties 1540
PostalCodeTaxRateData.PostalCode Property 1540
PostalCodeTaxRateData.RegionId Property 1540
PricingData Class 1540
PricingData Constructor 1541
PricingData.PricingData Constructor () 1541
PricingData.PricingData Constructor (List<CurrencyPricingData>) 1541
PricingData.PricingData Constructor (long, decimal, decimal) 1541
PricingData Methods 1542
PricingData.GetCurrencyById Method 1542
PricingData.GetCurrencyByIso Method 1542
PricingData.GetCurrencyByIsoNumber Method 1543
PricingData.GetListPrice Method 1543
PricingData.GetListPriceByCurrencyId Method 1543
PricingData.GetListPriceByIsoNumber Method 1543
PricingData.GetSalePrice Method 1543
PricingData.GetSalePriceByCurrencyId Method 1544
PricingData.GetSalePriceByIsoNumber Method 1544
PricingData.Validate Method 1544
PricingData Properties 1544
PricingData.CurrencyPricelist Property 1545
PricingData.IsRecurringPrice Property 1545
PricingData.Recurrence Property 1545

ProductData Class 1545
ProductData Constructor 1545
ProductData.ProductData Constructor () 1546
ProductData.ProductData Constructor (int) 1546
ProductData Properties 1546
ProductData.Variants Property 1546
ProductTypeAttributeData Class 1546
ProductTypeAttributeData Constructor 1547
ProductTypeAttributeData.ProductTypeAttributeData Constructor () 1547
ProductTypeAttributeData.ProductTypeAttributeData Constructor (string, int, EkEnumeration.ProductTypeAttributeDataType, object, EkEnumeration.ProductTypeAttributeStatus) 1547
ProductTypeAttributeData Properties 1548
ProductTypeAttributeData.ActiveStatus Property 1548
ProductTypeAttributeData.DataType Property 1548
ProductTypeAttributeData.DefaultValue Property 1548
ProductTypeAttributeData.DisplayOrder Property 1548
ProductTypeAttributeData.Id Property 1549
ProductTypeAttributeData.Name Property 1549
ProductTypeData Class 1549
ProductTypeData Properties 1549
ProductTypeData.Attributes Property 1550
ProductTypeData.DefaultThumbnails Property 1550
ProductTypeData.EntryClass Property 1550
RecommendationItemData Class 1550
RecommendationItemData Constructor 1551
RecommendationItemData.RecommendationItemData Constructor () 1551
RecommendationItemData.RecommendationItemData Constructor (long, long, long) 1551
RecommendationItemData.RecommendationItemData Constructor (long, long, long, EkEnumeration.RecommendationType) 1551
RecommendationItemData Properties 1552
RecommendationItemData.EntryLanguage Property 1552
RecommendationItemData.EntryType Property 1552
RecommendationItemData.Html Property 1552
RecommendationItemData.Image Property 1553
RecommendationItemData.ImageThumbnail Property 1553
RecommendationItemData.Pricing Property 1553
RecommendationItemData.Quicklink Property 1553
RecommendationItemData.Summary Property 1553
RecommendationItemData.Title Property 1553

RecommendationItemData.Type Property 1554
RegionData Class 1554
RegionData Constructor 1554
RegionData.RegionData Constructor () 1555
RegionData.RegionData Constructor (string, long, string, bool) 1555
RegionData Methods 1555
RegionData.ToString Method 1555
RegionData.Validate Method 1555
RegionData Properties 1556
RegionData.Code Property 1556
RegionData.CountryId Property 1556
RegionData.Enabled Property 1556
RegionData.Id Property 1557
RegionData.Name Property 1557
RegionTaxRateData Class 1557
RegionTaxRateData Constructor 1557
RegionTaxRateData.RegionTaxRateData Constructor () 1558
RegionTaxRateData.RegionTaxRateData Constructor (long, long, decimal) 1558
RegionTaxRateData.RegionTaxRateData Constructor (long, long, decimal, long) 1558
RegionTaxRateData Properties 1558
RegionTaxRateData.RegionData Property 1558
ShippingBaseData Class 1559
ShippingBaseData.ShippingBaseData Constructor 1559
ShippingBaseData Properties 1559
ShippingBaseData.Id Property 1559
ShippingBaseData.Name Property 1560
ShippingBaseData.ProviderService Property 1560
ShippingOptionData Class 1560
ShippingOptionData.ShippingOptionData Constructor 1560
ShippingOptionData Properties 1560
ShippingOptionData.CurrencyId Property 1561
ShippingOptionData.ShippingFee Property 1561
TaxClassData Class 1561
TaxClassData Constructor 1561
TaxClassData.TaxClassData Constructor () 1562
TaxClassData.TaxClassData Constructor (string) 1562
TaxClassData.TaxClassData Constructor (string, long) 1562
TaxClassData Properties 1562
TaxClassData.Id Property 1562

TaxClassData.Name Property 1563
TaxRateData Class 1563
TaxRateData Constructor 1563
TaxRateData.TaxRateData Constructor () 1564
TaxRateData.TaxRateData Constructor (long, long, long, decimal) 1564
TaxRateData.TaxRateData Constructor (long, long, long, decimal, long) 1564
TaxRateData Properties 1564
TaxRateData.Id Property 1564
TaxRateData.Rate Property 1565
TaxRateData.TaxClassId Property 1565
TaxRateData.TypeId Property 1565
TaxRateData.TypeItemId Property 1565
TaxTypeData Class 1565
TaxTypeData Constructor 1566
TaxTypeData.TaxTypeData Constructor () 1566
TaxTypeData.TaxTypeData Constructor (string, int) 1566
TaxTypeData.TaxTypeData Constructor (string, int, long) 1566
TaxTypeData Properties 1567
TaxTypeData.Id Property 1567
TaxTypeData.Name Property 1567
TaxTypeData.Precedence Property 1567
ThumbnailData Class 1568
ThumbnailData Properties 1568
ThumbnailData.Height Property 1568
ThumbnailData.Width Property 1568
ThumbnailDefaultData Class 1569
ThumbnailDefaultData Properties 1569
ThumbnailDefaultData.FileNameFormat Property 1569
ThumbnailDefaultData.Height Property 1569
ThumbnailDefaultData.Id Property 1570
ThumbnailDefaultData.MimeType Property 1570
ThumbnailDefaultData.Title Property 1570
ThumbnailDefaultData.Width Property 1570
TierPriceData Class 1570
TierPriceData Constructor 1571
TierPriceData.TierPriceData Constructor () 1571
TierPriceData.TierPriceData Constructor (int, decimal) 1571
TierPriceData.TierPriceData Constructor (int, decimal, long) 1572
TierPriceData Methods 1572

TierPriceData.Clone Method 1572
TierPriceData.CompareTo Method 1572
TierPriceData.Validate Method 1572
TierPriceData Properties 1573
TierPriceData.DataState Property 1573
TierPriceData.Id Property 1573
TierPriceData.Quantity Property 1573
TierPriceData.RecurrenceId Property 1573
TierPriceData.SalePrice Property 1574
ProductCouponEntryData Class 1574
ProductCouponEntryData Constructor 1574
ProductCouponEntryData.ProductCouponEntryData Constructor () 1575
ProductCouponEntryData.ProductCouponEntryData Constructor (long, EkEnumeration.CMSObjectTypes, long, string, bool, EkEnumeration.CatalogEntryType) 1575
ProductCouponEntryData Properties 1575
ProductCouponEntryData.EntryType Property 1575
WarehouseData Class 1575
WarehouseData Constructor 1576
WarehouseData.WarehouseData Constructor () 1576
WarehouseData.WarehouseData Constructor (string, AddressData) 1576
WarehouseData Properties 1576
WarehouseData.Address Property 1577
WarehouseData.Id Property 1577
WarehouseData.IsDefaultWarehouse Property 1577
WarehouseData.Name Property 1577
Weight Class 1577
Weight Constructor 1578
Weight.Weight Constructor () 1578
Weight.Weight Constructor (WeightUnit, float) 1578
Weight Methods 1578
Weight.Clone Method 1579
Weight.CompareTo Method 1579
Weight.ConvertUnits Method 1579
Weight Properties 1579
Weight.Amount Property 1579
Weight.Units Property 1580
ShippingMethodData Class 1580
ShippingMethodData.ShippingMethodData Constructor 1580
ShippingMethodData Properties 1580

ShippingMethodData.DisplayOrder Property 1580
ShippingMethodData.IsActive Property 1581
Structs, Records, Enums 1581
Ektron.Cms.Commerce.AddressProperty Enumeration 1582
Ektron.Cms.Commerce.CountryProperty Enumeration 1582
Ektron.Cms.Commerce.CouponProperty Enumeration 1582
Ektron.Cms.Commerce.CurrencyProperty Enumeration 1583
Ektron.Cms.Commerce.CustomerProperty Enumeration 1583
Ektron.Cms.Commerce.EntryAttributeProperty Enumeration 1583
Ektron.Cms.Commerce.EntryGroupItemProperty Enumeration 1583
Ektron.Cms.Commerce.EntryGroupProperty Enumeration 1584
Ektron.Cms.Commerce.EntryProperty Enumeration 1584
Ektron.Cms.Commerce.ExchangeRateProperty Enumeration 1584
Ektron.Cms.Commerce.LinearUnit Enumeration 1585
Ektron.Cms.Commerce.OrderProperty Enumeration 1585
Ektron.Cms.Commerce.PackageProperty Enumeration 1585
Ektron.Cms.Commerce.ProductTypeAttributeProperty Enumeration 1586
Ektron.Cms.Commerce.ProductTypeProperty Enumeration 1586
Ektron.Cms.Commerce.RegionProperty Enumeration 1586
Ektron.Cms.Commerce.ShippingMethodProperty Enumeration 1586
Ektron.Cms.Commerce.TaxClassProperty Enumeration 1587
Ektron.Cms.Commerce.TaxRateProperty Enumeration 1587
Ektron.Cms.Commerce.TaxRateType Enumeration 1587
Ektron.Cms.Commerce.TaxTypeProperty Enumeration 1588
Ektron.Cms.Commerce.WarehouseProperty Enumeration 1588
Ektron.Cms.Commerce.WeightUnit Enumeration 1588

Ektron.Cms.Common Namespace 1589
Classes 1589
CommerceSettings Class 1589
CommerceSettings Properties 1590
CommerceSettings.ComplianceMode Property 1590
CommerceSettings.CurrencyAlphaISO Property 1590
CommerceSettings.CurrencyCultureCode Property 1590
CommerceSettings.CurrencyId Property 1591
CommerceSettings.CurrencySymbol Property 1591
CommerceSettings.DefaultCurrencyId Property 1591
CommerceSettings.ISOCurrencySymbol Property 1591
CommerceSettings.OrderProcessingDisabled Property 1592
CommerceSettings.PasswordHistory Property 1592

CommerceSettings.TestMode Property 1592
UrlAliasAutoData Class 1592
UrlAliasAutoData Properties 1593
UrlAliasAutoData.AliasName Property 1593
UrlAliasAutoData.AutoAliasType Property 1593
UrlAliasAutoData.AutoId Property 1594
UrlAliasAutoData.Example Property 1594
UrlAliasAutoData.ExcludedPath Property 1594
UrlAliasAutoData.FileExtension Property 1594
UrlAliasAutoData.FormattedSelectedPath Property 1595
UrlAliasAutoData.HostName Property 1595
UrlAliasAutoData.IsEnabled Property 1595
UrlAliasAutoData.LanguageId Property 1595
UrlAliasAutoData.PageNameType Property 1595
UrlAliasAutoData.ReplacementCharacter Property 1596
UrlAliasAutoData.SelectedPath Property 1596
UrlAliasAutoData.SiteId Property 1596
UrlAliasAutoData.SourceId Property 1596
UrlAliasAutoData.SourceName Property 1597
UrlAliasAutoData.Target Property 1597
UrlAliasInfo Class 1597
UrlAliasInfo Properties 1597
UrlAliasInfo.AliasId Property 1598
UrlAliasInfo.CurrentAliasExtension Property 1598
UrlAliasInfo.CurrentAliasName Property 1598
UrlAliasInfo.PreviousAliasExtension Property 1598
UrlAliasInfo.PreviousAliasName Property 1599
UrlAliasManualData Class 1599
UrlAliasManualData Properties 1599
UrlAliasManualData.AliasId Property 1600
UrlAliasManualData.AliasName Property 1600
UrlAliasManualData.AliasPageName Property 1600
UrlAliasManualData.ContentId Property 1600
UrlAliasManualData.ContentLanguage Property 1601
UrlAliasManualData.ContentTitle Property 1601
UrlAliasManualData.DisplayAlias Property 1601
UrlAliasManualData.FileExtension Property 1601
UrlAliasManualData.HostName Property 1602
UrlAliasManualData.IsDefault Property 1602

UrlAliasManualData.IsEnabled Property 1602
UrlAliasManualData.LibraryId Property 1602
UrlAliasManualData.QueryString Property 1603
UrlAliasManualData.QueryStringAction Property 1603
UrlAliasManualData.SiteId Property 1603
UrlAliasManualData.Target Property 1603
UrlAliasRegExData Class 1604
UrlAliasRegExData Properties 1604
UrlAliasRegExData.Expression Property 1604
UrlAliasRegExData.ExpressionMap Property 1605
UrlAliasRegExData.ExpressionName Property 1605
UrlAliasRegExData.IsEnabled Property 1605
UrlAliasRegExData.Priority Property 1605
UrlAliasRegExData.RegExId Property 1606
UrlAliasRegExData.SiteId Property 1606
UrlAliasRegExData.TransformedUrl Property 1606

AssetUpdateData Class 1606
AssetUpdateData Properties 1607
AssetUpdateData.AssetId Property 1607
AssetUpdateData.Comment Property 1607
AssetUpdateData.ContentId Property 1608
AssetUpdateData.EndDate Property 1608
AssetUpdateData.EndDateAction Property 1608
AssetUpdateData.FileName Property 1609
AssetUpdateData.FolderId Property 1609
AssetUpdateData.GoLive Property 1609
AssetUpdateData.LanguageId Property 1609
AssetUpdateData.MetaData Property 1610
AssetUpdateData.TaxonomyTreeIds Property 1610
AssetUpdateData.Teaser Property 1610
AssetUpdateData.Title Property 1610

ContentRatingLabels Class 1611
ContentRatingLabels Properties 1611
ContentRatingLabels.AlreadyRatedMessage Property 1611
ContentRatingLabels.BadLabel Property 1612
ContentRatingLabels.CharsRemainingLabel Property 1612
ContentRatingLabels.CookiesRequiredMessage Property 1612
ContentRatingLabels.FeedbackHeader Property 1612

ContentRatingLabels.GoodLabel Property 1613
ContentRatingLabels.RatingLevelLabel Property 1613
ContentRatingLabels.ThankYouMessage Property 1613
ContentRatingLabels.TotalRatingsLabel Property 1613
ContentRatingLabels.UserCommentsHeader Property 1613
ContentRatingLabels.ValidationMessage Property 1614

DiscussionCategory Class 1614
DiscussionCategory Properties 1614
DiscussionCategory.BoardID Property 1614
DiscussionCategory.CategoryID Property 1615
DiscussionCategory.Name Property 1615
DiscussionCategory.SortOrder Property 1615

FolderRequest Class 1615
FolderRequest Properties 1616
FolderRequest.BreakInheritButton Property 1617
FolderRequest.CategoryRequired Property 1618
FolderRequest.ContentSubAssignments Property 1618
FolderRequest.DomainProduction Property 1618
FolderRequest.DomainStaging Property 1618
FolderRequest.EnableReplication Property 1619
FolderRequest.FileDirectory Property 1619
FolderRequest.FlagDefIdList Property 1619
FolderRequest.FlagInherited Property 1620
FolderRequest.FlagInheritedFrom Property 1620
FolderRequest.FolderCfldAssignments Property 1620
FolderRequest.FolderDescription Property 1621
FolderRequest.FolderId Property 1621
FolderRequest.FolderName Property 1621
FolderRequest.FolderType Property 1621
FolderRequest.ImageDirectory Property 1622
FolderRequest.IsDomainFolder Property 1622
FolderRequest.MetaInherited Property 1622
FolderRequest.MetaInheritedFrom Property 1623
FolderRequest.ParentId Property 1623
FolderRequest.PublishActive Property 1623
FolderRequest.SiteMapPath Property 1624
FolderRequest.SiteMapPathInherit Property 1624
FolderRequest.StyleSheet Property 1624

FolderRequest.SubscriptionProperties Property 1625
FolderRequest.suppressNotification Property 1625
FolderRequest.TaxonomyIdList Property 1625
FolderRequest.TaxonomyInherited Property 1626
FolderRequest.TaxonomyInheritedFrom Property 1626
FolderRequest.TemplateFileName Property 1626
FolderRequest.XmlConfiguration Property 1627
FolderRequest.XmlInherited Property 1627

FormFieldDataItem Class 1627
FormFieldDataItem Properties 1628
FormFieldDataItem.DataValue Property 1628
FormFieldDataItem.FieldName Property 1628

FormFieldDatalist Class 1628
FormFieldDatalist Properties 1629
FormFieldDatalist.CaptionXPath Property 1629
FormFieldDatalist.DatalistItems Property 1629
FormFieldDatalist.DataNamespaces Property 1629
FormFieldDatalist.DataSelect Property 1630
FormFieldDatalist.DataSrc Property 1630
FormFieldDatalist.Language Property 1630
FormFieldDatalist.Name Property 1631
FormFieldDatalist.ValueXPath Property 1631

FormFieldDatalistItem Class 1631
FormFieldDatalistItem Properties 1631
FormFieldDatalistItem.DataValue Property 1632
FormFieldDatalistItem.DisplayValue Property 1632
FormFieldDatalistItem.IsDefault Property 1632
FormFieldDatalistItem.IsEnabled Property 1633

FormFieldDefinition Class 1633
FormFieldDefinition Properties 1633
FormFieldDefinition.Basetype Property 1634
FormFieldDefinition.ContentType Property 1634
FormFieldDefinition.DatalistName Property 1634
FormFieldDefinition.Datatype Property 1635
FormFieldDefinition.DisplayName Property 1635
FormFieldDefinition.FieldName Property 1635
FormFieldDefinition.Language Property 1636

FormFieldDefinition.XPath Property 1636

FormFieldList Class 1636

FormFieldList Properties 1637

FormFieldList.Datalists Property 1637

FormFieldList.Fields Property 1637

FormRequestData Class 1637

FormRequestData Properties 1638

FormRequestData.AutoFill Property 1638

FormRequestData.FieldList Property 1639

FormRequestData.Id Property 1639

FormRequestData.IsPoll Property 1639

FormRequestData.Language Property 1639

FormRequestData.PostBackData Property 1640

FormRequestData.PreFillFielddata Property 1640

FormRequestData.TagRequired Property 1640

PagingInfo Class 1641

PagingInfo Properties 1641

PagingInfo.CurrentPage Property 1641

PagingInfo.EndRow Property 1642

PagingInfo.RecordsPerPage Property 1642

PagingInfo.StartRow Property 1642

PagingInfo.TotalPages Property 1642

PagingInfo.TotalRecords Property 1642

TaxonomyRequest Class 1643

TaxonomyRequest Properties 1643

TaxonomyRequest.Depth Property 1644

TaxonomyRequest.FolderId Property 1644

TaxonomyRequest.IncludeItems Property 1644

TaxonomyRequest.Preview Property 1645

TaxonomyRequest.ReadCount Property 1645

TaxonomyRequest.TaxonomyIdList Property 1645

TaxonomyRequest.TaxonomyItemType Property 1645

TaxonomyRequest.TaxonomyPath Property 1646

UriFavorite Class 1646

UriFavorite Properties 1647

UriFavorite.FavoriteDesc Property 1647

UriFavorite.FavoriteId Property 1647

UriFavorite.FavoriteName Property 1647
UriFavorite.FavoriteUri Property 1648
UriFavorite.UserId Property 1648

WorkSpaceGroupRequest Class 1648
WorkSpaceGroupRequest Properties 1649
WorkSpaceGroupRequest.GroupId Property 1649

WorkSpaceUserRequest Class 1649
WorkSpaceUserRequest Properties 1649
WorkSpaceUserRequest.UserId Property 1650

Index

a

Welcome

Welcome to Ektron CMS400.NET API Online Help. This documentation helps you understand the Server Control API (see page 7), Plug-in Extension API (see page 552), .NET Assemblies (see page 600), Web Services API (see page 1083) and the Data Classes (see page 1423) associated with Ektron CMS400.NET. Understanding these APIs allows you to enhance and extend your Ektron CMS400.NET site.

- The Server Control API (see page 7) section explains the methods and properties available in code behind for Ektron's CMS400.NET Server Controls. Use these methods and properties when you want to customize the use of a server control beyond the drag and drop functionality provided.
- The Plug-in Extension API (see page 552) documentation details the methods and properties that are available when creating a custom plug-in. Plug-in Extensions expose event hooks in Ektron CMS400.NET. As a developer, you can utilize these event hooks in Visual Studio C# or VB.NET to create customized events for your site.
- The .NET Assemblies (see page 600) are used to access functionality typically provided in the Workarea. For example, adding content, deleting a user or retrieving information about a specific folder. By using the .NET Assemblies you can create custom functionality while keeping within the Ektron Inc. Business Rules.
- The Web Services API (see page 1083) exposes many of the same methods as the .NET Assemblies for use with SOAP over HTTP.
- The Providers API (see page 1406) section describes the methods and properties needed to create custom payment gateways, inventory and shipping providers in code behind for use with CMS400.NET.
- The Data Classes (see page 1423) section describes the properties that allow you to get or set information for each item in Ektron CMS400.NET. For example, the ContentData data class allows you to get or set information about content, such as the date created, content type and the folder that contains the content.

Note: Although the following api lists ByVal in several locations, Microsoft .net defaults to ByRef when one is not specified. This Microsoft Article explains further:

http://msdn.microsoft.com/en-us/library/ee478101(VS.85).aspx

How Do I...

Below is a list of actions that developers often want to accomplish.

How Do I...

Blogs

create a Web page from which a user can add a blog post? (see page 827)

create a Web page from which a user can retrieve blog information by specifying a blog ID, a date and how to handle post information? (see page 831)

Content

create a Web page that allows users to add content? (see page 843)

create a Web page that allows users to delete content by entering the content's ID? (see page 848)

create a Web page from which a user can view a content block and its title by entering the content ID into a text box?

ContentRating

create a Web page that retrieves a content rating element used to rate content? (see page 866)

Folder

create a Web page from which a folder can be added to the CMS400.NET? (see page 684)

create a Web page from which a folder can be deleted from CMS400.NET by entering its ID? (see page 685)

create a Web page from which a folder ID can be entered and list of child folders returned? (see page 688)

Library

create a Web page that displays a graphical list of all images stored in the library for a specific folder in the Workarea? (see page 697)

Metadata

create a Web page from which you can add a new metadata definition? (see page 715)

create a Web page from which all metadata for a specific content ID can be retrieved? (see page 717)

SiteMap

create a Web page from which the breadcrumb path for a given folder can be retrieved? (see page 732)

create a Web page that displays the site map for a given folder? (see page 734)

Threaded Discussions

create a Web page from which a forum can be added to a discussion board? (see page 884)

create a Web page from which forum data can be retrieved? (see page 893)

Users and Membership Users

create a Web page from which a new membership user can be added? (see page 961)

create a Web page that retrieves user details for a given username? (see page 974)

create a Web page that programmatically logs a user into CMS400.NET? (see page 988)

List of Developer Resources

List of Developer Resources

In addition to the information in this manual, the following resources provide information on developing your site. Below is a quick reference list of resources.

- Ektron's DevCenter
- http://dev.ektron.com
- Ektron CMS400.NET Developer Manual
- http://www.ektron.com/manuals/cms400/DevelopersManual.pdf
- Ektron's Developer Training Course
- http://www.ektron.com/cms400-web-cms.aspx?id=2653
- Developer Site Provided with Ektron CMS400.NET Developer Starter Site Install
- http://localhost/CMS400Developer/default.aspx
- Ektron Support and Product Pages
- http://www.ektron.com/support/cms400_support.aspx
- http://www.ektron.com/web-content-management-solutions.aspx
- Supporting Documents
- http://www.ektron.com/web-content-management-solutions.aspx?id=4020

Ektron's DevCenter

http://dev.ektron.com

The Ektron DevCenter includes a wide array of resources to assist our developers and partners. Whether you're looking for a solution to a specific problem or just seeing what you might be able to do with CMS400.NET, you should be able to find what you need here.

User Discussion Boards

http://dev.ektron.com/forum.aspx

Use the Forum to exchange ideas and information about Ektron, Inc. products.

Ektron Blogs

http://dev.ektron.com/blogs.aspx

With Blogs, learn what developers and engineers at Ektron, Inc. are thinking.

Ektron Fileshare

http://dev.ektron.com/fileshare.aspx

The Ektron FileShare is a system designed to allow developers to share server controls, code snippets, classes and other tools that may be useful to other developers.

Tutorials

http://dev.ektron.com/tutorials.aspx

This area provides video tutorials to help you learn how to use CMS400.

External Resources

http://dev.ektron.com/template.aspx?id=2756

The sites listed on this page contain tutorials, developer news, or other materials that might be useful to you while developing your CMS400.NET site. These sites are not run by Ektron, Inc.

Knowledge Base Articles

http://dev.ektron.com/kb_search.aspx

These articles provide information about all Ektron, Inc. products.

Ektron CMS400.NET Developer Manual

http://www.ektron.com/manuals/cms400/DevelopersManual.pdf

This manual can be used as a reference for the developer, or the person is who is setting up your Ektron CMS400.NET Web site.

Besides explaining how to insert the functions to your Web templates manually, this manual also explains how to create Ektron CMS400.NET templates in a WYSIWYG environment by using the Ektron supplied server controls for Visual Studio 2005 or by using the Dreamweaver extensions supplied with the installation.

Ektron's Developer Training Course

http://www.ektron.com/cms400-web-cms.aspx?id=2653

Why should you become a Certified Ektron Developer?

- Learn best practices for deploying an Ektron CMS400.NET driven Web site
- Learn how to extend CMS400.NET to do more of what you want
- See real-world examples of how an Ektron CMS is deployed
- Be an expert on the most commonly deployed CMS
- Add "Certified Ektron Developer" to your resume

Developer Site Provided with Ektron CMS400.NET Starter Site Install

http://localhost/CMS400Developer/default.aspx

Replace localhost with location of your site.

Ektron has compiled a list of several different examples of developing your Web site with Ektron CMS400.NET. These examples show how easy it is to create robust Web pages by using "out-of-the-box" functionality including server controls.

Ektron Support and Product Pages

http://www.ektron.com/support/cms400_support.aspx

http://www.ektron.com/web-content-management-solutions.aspx

Ektron's Support and Product pages are a great launch pad to learn about Ektron CMS400.NET.

Supporting Documents

You can find more detailed information about Ektron CMS400.NET in the following manuals:

Ektron CMS400.NET Administrator Manual

Ektron CMS400.NET User Manual

Ektron CMS400.NET Quick Start Manual

Ektron CMS400.NET Setup Manual

Ektron Namespace

Server Controls API

The Server Control API section explains the methods and properties available in code behind for Ektron's CMS400.NET Server Controls.

Ektron CMS400.NET's server controls let you insert many standard methods and properties within a Visual Studio environment. This means that you can see the effect of your changes in real time -- you don't have to modify a page then compile a sample project to see the results.

For additional information on the Ektron CMS400.NET Server Controls, see the Developer Manual section > "Introduction to Ektron CMS400.NET Server Controls".

Namespaces

Name	Description
Cms (see page 7)	The CMS namespace contains the Server Control classes.

Ektron.Cms Namespace

The CMS namespace contains the Server Control classes.

Namespaces

Name	Description
Controls (see page 7)	This is namespace Ektron.Cms.Controls.

Ektron.Cms.Controls Namespace

This is namespace Ektron.Cms.Controls.

Namespaces

Name	Description
CalendarProvider (see page 11)	This is namespace Ektron.Cms.Controls.CalendarProvider.
Commerce (see page 11)	This is namespace Ektron.Cms.Controls.Commerce.
ECommerce (see page 11)	This is namespace Ektron.Cms.Controls.ECommerce.
Personalization (see page 11)	This is namespace Ektron.Cms.Controls.Personalization.
EktronUserControl (see page 12)	This is namespace Ektron.Cms.Controls.EktronUserControl.
Support (see page 12)	This is namespace Ektron.Cms.Controls.Support.
WebCalendarForms (see page 548)	This is namespace Ektron.Cms.Controls.WebCalendarForms.

Classes

	Name	Description
	ActiveTopics (see page 15)	The ActiveTopics server control displays either the most active topics or the most recent topics. The most active are determined by how many new posts are added to a topic or how many replies a post receives. The most recent are determined by the date of the post.
	ActivityStream (see page 20)	This is class Ektron.Cms.Controls.ActivityStream.
	AnalyticsTracker (see page 25)	[deprecated]AnalyticsTracker server control has been deprecated. Instead use Analytic server control to track statistics about visits to your Web site.
	AssetControl (see page 27)	The AssetControl server control, when viewed on a Web form, displays a drag and drop box for users to upload assets to CMS400.NET or update an existing asset.
	Blog (see page 32)	The Blog server control allows you to quickly add a blog to a Web form. It has all of the items commonly found on a blog page in one server control.
	BlogArchive (see page 39)	The BlogArchive server control displays a list of past months or years that have posts.
	BlogCalendar (see page 42)	The BlogCalendar server control allows a user to display a calendar on a Web page and associate it with a blog.
	BlogCategories (see page 44)	The BlogCategories server control displays a blog's categories on a Web form.
	BlogEntries (see page 47)	The BlogEntries server control allows you to display a list of multiple blog posts on a Web form.
	BlogPost (see page 54)	The BlogPost server control allows you to display an individual blog post on a page.
	BlogRecentPosts (see page 56)	The BlogRecentPosts server control displays a list of recent blog posts links on a Web form.
	BlogRoll (see page 59)	The BlogRoll server control displays the blog roll for a blog on a Web form.
	BlogRSS (see page 61)	The BlogRSS server control displays the icon for the blog's RSS feed on the Web form.
	BreadCrumb (see page 63)	The BreadCrumb server control displays history breadcrumbs. Your site visitors can use the breadcrumb trail to jump to previously visited pages.
	BreadCrumbItem (see page 70)	Class to contain compass item data.
	BusinessRules (see page 73)	Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example: • **Condition** - a site visitor visits your site for the first time • **Action** - add a cookie to the site visitor's system and pop up a welcome message With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control... more (see page 73)
	Calendar (see page 76)	Summary description for Calendar.
	Cart (see page 81)	This is class Ektron.Cms.Controls.Cart.
	Checkout (see page 85)	This is class Ektron.Cms.Controls.Checkout.
	Collection (see page 95)	Ektron Collection class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.
	CommunityDocuments (see page 106)	This is class Ektron.Cms.Controls.CommunityDocuments.
	CommunityGroupBrowser (see page 111)	Summary description for Message board
	CommunityGroupList (see page 117)	Summary description for Message board
	CommunityGroupMembers (see page 124)	Summary description for Message board
	CommunityGroupProfile (see page 130)	
	CommunitySearch (see page 135)	This is class Ektron.Cms.Controls.CommunitySearch.
	ContentBlock (see page 148)	Summary description for ContentBlock.
	ContentFlagging (see page 155)	
	ContentList (see page 159)	The ContentList server control is used to display a list of content blocks on a Web page. In contrast to a List Summary, where content must be in a specified folder, the ContentList server control uses content from any CMS400.NET folder.

	ContentRating (see page 169)	Use the ContentRating server control to place a rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. A visitor can rank the content on a numerical scale of one through nine, as well as submit a text comment.
	ContentReview (see page 177)	The ContentReview server control has two ways it can be used. The first, places a star based rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. An item is rated when a visitor clicks on a star. Depending on the XSLT chosen, visitors can also submit review comments. The second way the ContentReview control can be used is to retrieve and display visitor reviews and comments. In this mode, the control can display reviews of a content item or reviews by... more (see page 177)
	CurrencySelect (see page 183)	Summary description for CurrencySelect control
	DesignTimeDiagnostic (see page 186)	The DesignTimeDiagnostic server control is used to verify the connection to Ektron's server controls Web service. When added to a Web form, this control provides the following information: • **Web Service Path** - displays the server control Web service path in your web.config file • **Web Service Exists** - calls a method in the Web service that returns True when the Web service exists. When it does not, it returns False. • **Test Login (see page 347) Control** - displays the Login (see page 347) server control to show that the Web service is connected and working properly. When not connecting properly, an error message is displayed.
	DhtmlMenu (see page 188)	The DhtmlMenu server control lets you display a menu with drop down boxes and submenus. This lets developers add a menu to a page without taking up a lot space.
	Directory (see page 196)	The Directory server control lets you customize the behavior of the Taxonomy (see page 876) feature. You place this control on any Web form to display a taxonomy.
	EktronCatalogPart (see page 210)	When placed inside a Microsoft CatalogZone, the EktronCatalogPart allows users to choose the type of CMS400.NET control that is added to a WebPartZone.
	EktronCommunityCatalogPart (see page 217)	The EktronCommunityCatalogPart Server Control gets a list of Discussion Board (Forums) and Blogs from your site. Then, it allows site users to choose from any available Discussion Boards and Blogs and add them to a WebPartZone.
	EktronWebPartZone (see page 219)	Similar to a Microsoft WebPartZone, the EktronWebPartZone server control provides a location for content to be displayed on a Web page. The difference between the two is formatting. The EktronWebPartZone is designed to work specifically with Ektron Server Controls, while Microsoft WebPartZones are more generic.
	EkWebControl (see page 220)	The base class from which all Ektron CMS400 Server Controls are derived.
	EkWebPart (see page 229)	This is class Ektron.Cms.Controls.EkWebPart.
	EkXsltBase (see page 232)	Summary description for EkXsltBase.
	ExplorerDragDrop (see page 234)	Defines the Explorer Drag Drop control, used to upload assets via Desktop drag drop to the CMS.
	Favorites (see page 236)	Summary description for Favorites
	FlexMenu (see page 242)	The FlexMenu server control displays a menu on a Web form. Although it looks and acts like a Smart Menu (see page 368), its architecture is substantially different. Specifically, a FlexMenu creates XML. So, you can modify its behavior using an XSLT file, and change its appearance using a cascading style sheet (.css) file.
	FolderBreadcrumb (see page 250)	The FolderBreadcrumb server control displays sitemap breadcrumbs. This type of trail is typically made up of the folder path for the content. It is important to note that the FolderBreadcrumb server control does not simply read your folder structure and display its path. It reads breadcrumb information set in the Workarea's folder properties for each folder. Administrators define a folder's breadcrumbs on the Breadcrumb tab located in the Workarea folder properties. *See Also:* The Administrator Manual > "Managing Content (see page 822)" > "Managing Content (see page 822) Folders" > "Breadcrumbs" For additional information on this server control, see the Developer Manual section "Introduction to... more (see page 250)
	FormBlock (see page 254)	The FormBlock server control displays a content block associated with a form.
	FormFieldItem (see page 261)	Defines a field within the form.
	Forum (see page 264)	The Forum server control displays a discussion board on a Web page.
	Friends (see page 279)	Summary description for Friends (Colleague Server Control API (see page 669))
	HtmlEditDisplay (see page 286)	This is class Ektron.Cms.Controls.HtmlEditDisplay.

	HtmlEditor (see page 289)	This server control provides the ability to edit specific CMS content. The eWebEditPro editor can easily be placed onto a page for content editing. There is flexibility to control the toolbars, the available functionality, and the operations of the editor. The control, at run time, produces the fields listed below. They are posted with the content and can be processed in the receiving page. • [ControlName]_Action = the action to take - save, check in, undo checkout • [ControlName]_ContentTitle = the content title • [ControlName]_ContentId = the ID of the content • [ControlName]_Language = the language of the content The content title caption is... more (see page 289)
	ImageControl (see page 305)	The ImageControl server control, when viewed on a Web form, displays an image stored within the CMS400.NET document management feature. When a user is logged in and has permission to edit the image, he can right click the image and select edit. This creates a drag and drop box that the user can use to update the file.
	IndexSearch (see page 308)	Use the IndexSearch server control to display a Smart Form (see page 871)'s search screen on any Web form. This lets a user search your site based on XML parameters.
	Invite (see page 325)	
	LanguageAPI (see page 328)	Control that can be used to override the default system language assignment behavior.
	LanguageSelect (see page 333)	The LanguageSelect server control displays a language selection dropdown list on an Ektron CMS400.NET Web page. The dropdown box lists all languages selected on the Settings > Configuration > Language (see page 227) settings screen.
	ListSummary (see page 335)	Summary description for ListSummary.
	Login (see page 347)	Summary description for Login.
	Map (see page 350)	Map control
	Membership (see page 360)	The Membership server control creates a form on a Web page that allows a user to create or update a profile of themselves in your membership list. The control creates the form based on the standard Ektron user properties and any custom user properties that exist. It also provides a way to have new membership users read and accept the terms and conditions for using Discussion Boards (Forums) when needed.
	Menu (see page 368)	The Menu server control calls a menu to be displayed on a page. Using the Menu server control, you can manipulate a menu by using the DisplayXslt (see page 370) property.
	MessageBoard (see page 371)	Summary description for Message board
	Messaging (see page 380)	
	MetaData (see page 386)	The Metadata server control lets you add the metadata from content blocks to a Web page. This lets developers add metadata quickly without having to type it in. You can add metadata from a single content block, multiple content blocks, or by dynamically passing a content ID from a URL.
	MetaDataList (see page 390)	Ektron ContentByMeta class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.
	MicroMessaging (see page 401)	Summary description for MicroMessage (see page 1070)
	MyAccount (see page 409)	This is class Ektron.Cms.Controls.MyAccount.
	OrderList (see page 414)	Summary description for OrderList control
	PersonalizationManager (see page 419)	The PersonalizationManager server control allows site users to personalize, add, edit and hide content on a Web page. The level of privileges assigned to a site user in CMS400.NET determines what personalization functionality is available when the control is active. The Personalization (see page 11) manager only appears when a user is logged in to the site.
	PhotoGallery (see page 425)	This is class Ektron.Cms.Controls.PhotoGallery.
	Poll (see page 425)	The Poll Server Control displays a poll or survey created from an Ektron CMS400.NET form on a Web page.
	PostHistory (see page 428)	The PostHistory server control displays a list of posts for a given user. The posts are displayed in order by date and contain the following: Topic - the topic that contains the post Posted - the date the post was created Content (see page 822) - the content with in the post
	PostLoginEventArgs (see page 431)	Holds the Post-Login (see page 347) event arguments.
	Product (see page 431)	Summary description for Product
	ProductList (see page 437)	This is class Ektron.Cms.Controls.ProductList.
	ProductSearch (see page 445)	The Web Search (see page 474) server control lets you customize the behavior of the search. You place this control on any Web form from which a site visitor can search your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 725)" > "Web Site (see page 725) Search (see page 474)."
	Recommendation (see page 462)	

	RssAggregator (see page 468)	Ektron RssAggregator class. Allows access to a RSS Feed object through IEnumerator, ICollection and IList interfaces.
	RssItemBase (see page 473)	Container for Title (see page 474), Link (see page 474), Description (see page 474) and DatePosted (see page 474) article information.
	Search (see page 474)	Search object returns the control and results.
	Sitemap (see page 496)	The Sitemap server control utilizes the folder breadcrumb information in the Workarea to display a sitemap of your site. By choosing the starting point of the sitemap, the max levels to display and applying a style class, you can customize the sitemap.
	SmartMenu (see page 500)	Class extends the Menu (see page 368)-Sever-Control, adding CSS based menus.
	SocialBar (see page 507)	Summary description for ListSummary (see page 335).
	TagCloud (see page 513)	The TagCloud server control allows you to visually depict a weighted list of tags assigned to users or community groups. The weight of each tag is determined by the amount of times a tag is assigned and represented by font size. The large the font size the more times a tag is assigned. Tags (see page 805) are keywords that are associated with a user or community group in CMS400.NET.
	UserProfile (see page 520)	
	WebCalendar (see page 525)	Summary description for Calendar (see page 76).
	WebSearch (see page 529)	The Web Search (see page 474) server control lets you customize the behavior of the search. You place this control on any Web form from which a site visitor can search your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 725)" > "Web Site (see page 725) Search (see page 474)."

Structs, Records, Enums

	Name	Description
	CommunityDocumentsObjectType (see page 548)	This control allows user / groups to manage their CommunityDocuments (see page 106).

Ektron.Cms.Controls.CalendarProvider Namespace

This is namespace Ektron.Cms.Controls.CalendarProvider.

Ektron.Cms.Controls.Commerce Namespace

This is namespace Ektron.Cms.Controls.Commerce.

Ektron.Cms.Controls.ECommerce Namespace

This is namespace Ektron.Cms.Controls.ECommerce.

Ektron.Cms.Controls.Personalization Namespace

This is namespace Ektron.Cms.Controls.Personalization.

Classes

	Name	Description
	EkEditorPart (see page 12)	This is class Ektron.Cms.Controls.Personalization.EkEditorPart.

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	EkEditorPart (see page 12)	This is class Ektron.Cms.Controls.Personalization.EkEditorPart.

EkEditorPart Class

Class Hierarchy

C#

```
internal class EkEditorPart : EditorPart;
```

Visual Basic

```
Friend Class EkEditorPart
Inherits EditorPart
```

File

EkEditorPart.cs

Description

This is class Ektron.Cms.Controls.Personalization.EkEditorPart.

Ektron.Cms.Controls.EktronUserControl Namespace

This is namespace Ektron.Cms.Controls.EktronUserControl.

Ektron.Cms.Controls.Support Namespace

This is namespace Ektron.Cms.Controls.Support.

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	ActiveTopics (see page 15)	The ActiveTopics server control displays either the most active topics or the most recent topics. The most active are determined by how many new posts are added to a topic or how many replies a post receives. The most recent are determined by the date of the post.
	ActivityStream (see page 20)	This is class Ektron.Cms.Controls.ActivityStream.
	AnalyticsTracker (see page 25)	[deprecated]AnalyticsTracker server control has been deprecated. Instead use Analytic server control to track statistics about visits to your Web site.
	AssetControl (see page 27)	The AssetControl server control, when viewed on a Web form, displays a drag and drop box for users to upload assets to CMS400.NET or update an existing asset.
	Blog (see page 32)	The Blog server control allows you to quickly add a blog to a Web form. It has all of the items commonly found on a blog page in one server control.
	BlogArchive (see page 39)	The BlogArchive server control displays a list of past months or years that have posts.
	BlogCalendar (see page 42)	The BlogCalendar server control allows a user to display a calendar on a Web page and associate it with a blog.
	BlogCategories (see page 44)	The BlogCategories server control displays a blog's categories on a Web form.
	BlogEntries (see page 47)	The BlogEntries server control allows you to display a list of multiple blog posts on a Web form.
	BlogPost (see page 54)	The BlogPost server control allows you to display an individual blog post on a page.
	BlogRecentPosts (see page 56)	The BlogRecentPosts server control displays a list of recent blog posts links on a Web form.
	BlogRoll (see page 59)	The BlogRoll server control displays the blog roll for a blog on a Web form.
	BlogRSS (see page 61)	The BlogRSS server control displays the icon for the blog's RSS feed on the Web form.
	BreadCrumb (see page 63)	The BreadCrumb server control displays history breadcrumbs. Your site visitors can use the breadcrumb trail to jump to previously visited pages.

	BreadCrumbItem (see page 70)	Class to contain compass item data.
	BusinessRules (see page 73)	Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example: • **Condition** - a site visitor visits your site for the first time • **Action** - add a cookie to the site visitor's system and pop up a welcome message With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control... more (see page 73)
	Calendar (see page 76)	Summary description for Calendar.
	Cart (see page 81)	This is class Ektron.Cms.Controls.Cart.
	Checkout (see page 85)	This is class Ektron.Cms.Controls.Checkout.
	Collection (see page 95)	Ektron Collection class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.
	CommunityDocuments (see page 106)	This is class Ektron.Cms.Controls.CommunityDocuments.
	CommunityGroupBrowser (see page 111)	Summary description for Message board
	CommunityGroupList (see page 117)	Summary description for Message board
	CommunityGroupMembers (see page 124)	Summary description for Message board
	CommunityGroupProfile (see page 130)	
	CommunitySearch (see page 135)	This is class Ektron.Cms.Controls.CommunitySearch.
	ContentBlock (see page 148)	Summary description for ContentBlock.
	ContentFlagging (see page 155)	
	ContentList (see page 159)	The ContentList server control is used to display a list of content blocks on a Web page. In contrast to a List Summary, where content must be in a specified folder, the ContentList server control uses content from any CMS400.NET folder.
	ContentRating (see page 169)	Use the ContentRating server control to place a rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. A visitor can rank the content on a numerical scale of one through nine, as well as submit a text comment.
	ContentReview (see page 177)	The ContentReview server control has two ways it can be used. The first, places a star based rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. An item is rated when a visitor clicks on a star. Depending on the XSLT chosen, visitors can also submit review comments. The second way the ContentReview control can be used is to retrieve and display visitor reviews and comments. In this mode, the control can display reviews of a content item or reviews by... more (see page 177)
	CurrencySelect (see page 183)	Summary description for CurrencySelect control
	DesignTimeDiagnostic (see page 186)	The DesignTimeDiagnostic server control is used to verify the connection to Ektron's server controls Web service. When added to a Web form, this control provides the following information: • **Web Service Path** - displays the server control Web service path in your web.config file • **Web Service Exists** - calls a method in the Web service that returns True when the Web service exists. When it does not, it returns False. • **Test Login (see page 347) Control** - displays the Login (see page 347) server control to show that the Web service is connected and working properly. When not connecting properly, an error message is displayed.
	DhtmlMenu (see page 188)	The DhtmlMenu server control lets you display a menu with drop down boxes and submenus. This lets developers add a menu to a page without taking up a lot space.
	Directory (see page 196)	The Directory server control lets you customize the behavior of the Taxonomy (see page 876) feature. You place this control on any Web form to display a taxonomy.
	EktronCatalogPart (see page 210)	When placed inside a Microsoft CatalogZone, the EktronCatalogPart allows users to choose the type of CMS400.NET control that is added to a WebPartZone.

	EktronCommunityCatalogPart (see page 217)	The EktronCommunityCatalogPart Server Control gets a list of Discussion Board (Forums) and Blogs from your site. Then, it allows site users to choose from any available Discussion Boards and Blogs and add them to a WebPartZone.
	EktronWebPartZone (see page 219)	Similar to a Microsoft WebPartZone, the EktronWebPartZone server control provides a location for content to be displayed on a Web page. The difference between the two is formatting. The EktronWebPartZone is designed to work specifically with Ektron Server Controls (see page 7), while Microsoft WebPartZones are more generic.
	EkWebControl (see page 220)	The base class from which all Ektron CMS400 Server Controls (see page 7) are derived.
	EkWebPart (see page 229)	This is class Ektron.Cms.Controls.EkWebPart.
	EkXsltBase (see page 232)	Summary description for EkXsltBase.
	ExplorerDragDrop (see page 234)	Defines the Explorer Drag Drop control, used to upload assets via Desktop drag drop to the CMS.
	Favorites (see page 236)	Summary description for Favorites
	FlexMenu (see page 242)	The FlexMenu server control displays a menu on a Web form. Although it looks and acts like a Smart Menu (see page 368), its architecture is substantially different. Specifically, a FlexMenu creates XML. So, you can modify its behavior using an XSLT file, and change its appearance using a cascading style sheet (.css) file.
	FolderBreadcrumb (see page 250)	The FolderBreadcrumb server control displays sitemap breadcrumbs. This type of trail is typically made up of the folder path for the content. It is important to note that the FolderBreadcrumb server control does not simply read your folder structure and display its path. It reads breadcrumb information set in the Workarea's folder properties for each folder. Administrators define a folder's breadcrumbs on the Breadcrumb tab located in the Workarea folder properties. *See Also:* The Administrator Manual > "Managing Content (see page 822)" > "Managing Content (see page 822) Folders" > "Breadcrumbs" For additional information on this server control, see the Developer Manual section "Introduction to... more (see page 250)
	FormBlock (see page 254)	The FormBlock server control displays a content block associated with a form.
	FormFieldItem (see page 261)	Defines a field within the form.
	Forum (see page 264)	The Forum server control displays a discussion board on a Web page.
	Friends (see page 279)	Summary description for Friends (Colleague Server Control API (see page 669))
	HtmlEditDisplay (see page 286)	This is class Ektron.Cms.Controls.HtmlEditDisplay.
	HtmlEditor (see page 289)	This server control provides the ability to edit specific CMS content. The eWebEditPro editor can easily be placed onto a page for content editing. There is flexibility to control the toolbars, the available functionality, and the operations of the editor. The control, at run time, produces the fields listed below. They are posted with the content and can be processed in the receiving page. • [ControlName]_Action = the action to take - save, check in, undo checkout • [ControlName]_ContentTitle = the content title • [ControlName]_ContentId = the ID of the content • [ControlName]_Language = the language of the content The content title caption is... more (see page 289)
	ImageControl (see page 305)	The ImageControl server control, when viewed on a Web form, displays an image stored within the CMS400.NET document management feature. When a user is logged in and has permission to edit the image, he can right click the image and select edit. This creates a drag and drop box that the user can use to update the file.
	IndexSearch (see page 308)	Use the IndexSearch server control to display a Smart Form (see page 871)'s search screen on any Web form. This lets a user search your site based on XML parameters.
	Invite (see page 325)	
	LanguageAPI (see page 328)	Control that can be used to override the default system language assignment behavior.
	LanguageSelect (see page 333)	The LanguageSelect server control displays a language selection dropdown list on an Ektron CMS400.NET Web page. The dropdown box lists all languages selected on the Settings > Configuration > Language (see page 227) settings screen.
	ListSummary (see page 335)	Summary description for ListSummary.
	Login (see page 347)	Summary description for Login.
	Map (see page 350)	Map control
	Membership (see page 360)	The Membership server control creates a form on a Web page that allows a user to create or update a profile of themselves in your membership list. The control creates the form based on the standard Ektron user properties and any custom user properties that exist. It also provides a way to have new membership users read and accept the terms and conditions for using Discussion Boards (Forums) when needed.
	Menu (see page 368)	The Menu server control calls a menu to be displayed on a page. Using the Menu server control, you can manipulate a menu by using the DisplayXslt (see page 370) property.

	MessageBoard (see page 371)	Summary description for Message board
	Messaging (see page 380)	
	MetaData (see page 386)	The Metadata server control lets you add the metadata from content blocks to a Web page. This lets developers add metadata quickly without having to type it in. You can add metadata from a single content block, multiple content blocks, or by dynamically passing a content ID from a URL.
	MetaDataList (see page 390)	Ektron ContentByMeta class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.
	MicroMessaging (see page 401)	Summary description for MicroMessage (see page 1070)
	MyAccount (see page 409)	This is class Ektron.Cms.Controls.MyAccount.
	OrderList (see page 414)	Summary description for OrderList control
	PersonalizationManager (see page 419)	The PersonalizationManager server control allows site users to personalize, add, edit and hide content on a Web page. The level of privileges assigned to a site user in CMS400.NET determines what personalization functionality is available when the control is active. The Personalization (see page 11) manager only appears when a user is logged in to the site.
	PhotoGallery (see page 425)	This is class Ektron.Cms.Controls.PhotoGallery.
	Poll (see page 425)	The Poll Server Control displays a poll or survey created from an Ektron CMS400.NET form on a Web page.
	PostHistory (see page 428)	The PostHistory server control displays a list of posts for a given user. The posts are displayed in order by date and contain the following: Topic - the topic that contains the post Posted - the date the post was created Content (see page 822) - the content with in the post
	PostLoginEventArgs (see page 431)	Holds the Post-Login (see page 347) event arguments.
	Product (see page 431)	Summary description for Product
	ProductList (see page 437)	This is class Ektron.Cms.Controls.ProductList.
	ProductSearch (see page 445)	The Web Search (see page 474) server control lets you customize the behavior of the search. You place this control on any Web form from which a site visitor can search your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 725)" > "Web Site (see page 725) Search (see page 474)."
	Recommendation (see page 462)	
	RssAggregator (see page 468)	Ektron RssAggregator class. Allows access to a RSS Feed object through IEnumerator, ICollection and IList interfaces.
	RssItemBase (see page 473)	Container for Title (see page 474), Link (see page 474), Description (see page 474) and DatePosted (see page 474) article information.
	Search (see page 474)	Search object returns the control and results.
	Sitemap (see page 496)	The Sitemap server control utilizes the folder breadcrumb information in the Workarea to display a sitemap of your site. By choosing the starting point of the sitemap, the max levels to display and applying a style class, you can customize the sitemap.
	SmartMenu (see page 500)	Class extends the Menu (see page 368)-Sever-Control, adding CSS based menus.
	SocialBar (see page 507)	Summary description for ListSummary (see page 335).
	TagCloud (see page 513)	The TagCloud server control allows you to visually depict a weighted list of tags assigned to users or community groups. The weight of each tag is determined by the amount of times a tag is assigned and represented by font size. The large the font size the more times a tag is assigned. Tags (see page 805) are keywords that are associated with a user or community group in CMS400.NET.
	UserProfile (see page 520)	
	WebCalendar (see page 525)	Summary description for Calendar (see page 76).
	WebSearch (see page 529)	The Web Search (see page 474) server control lets you customize the behavior of the search. You place this control on any Web form from which a site visitor can search your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 725)" > "Web Site (see page 725) Search (see page 474)."

ActiveTopics Class

The ActiveTopics server control displays either the most active topics or the most recent topics. The most active are determined by how many new posts are added to a topic or how many replies a post receives. The most recent are determined by the date of the post.

Class Hierarchy

C#

```
[DefaultProperty("BoardID"), ToolboxData("<{0}:ActiveTopics runat=server></{0}:ActiveTopics>")]
public class ActiveTopics : EkWebControl;
```

Visual Basic

```
<DefaultProperty("BoardID"), ToolboxData("<{0}:ActiveTopics runat=server></{0}:ActiveTopics>")>
Public Class ActiveTopics
Inherits EkWebControl
```

File

ActiveTopics.cs

ActiveTopics Methods

The methods of the ActiveTopics class are listed here.

Public Methods

	Name	Description
	Fill (see page 16)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 16)	Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.
	Wrap (see page 16)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

ActiveTopics.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

ActiveTopics.ToString Method

Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

ActiveTopics.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

ActiveTopics Properties

The properties of the ActiveTopics class are listed here.

Public Properties

	Name	Description
	BoardID (see page 17)	The ID of the board from which to get entries.
	CacheInterval (see page 17)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, set the CacheInterval property to 300 (three hundred).
	DisplayMode (see page 18)	Choose between Active or Recent. In Active mode, the server control displays the most active posts. In Recent mode, the server control displays the latest posts. The default is Active.
	DynamicForumParameter (see page 18)	Gets or Sets the QueryString parameter to read a Forum (see page 264) ID dynamically.
	DynamicGroupParameter (see page 18)	Gets or Sets the QueryString parameter to read a Forum (see page 264) ID dynamically.
	DynamicThreadParameter (see page 19)	Gets or Sets the QueryString parameter to read a Forum (see page 264) ID dynamically.
	MaxNumber (see page 19)	The maximum number of topics listed. The default is 10.
	URLPath (see page 19)	The URL path to the Forum (see page 264) server control's Web page. For example:

ActiveTopics.BoardID Property

The ID of the board from which to get entries.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the board from which to get entries.")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BoardID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the board from which to get entries.")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BoardID() As long
```

ActiveTopics.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
Public virtual Property CacheInterval() As double
```

ActiveTopics.DisplayMode Property

Choose between Active or Recent. In Active mode, the server control displays the most active posts. In Recent mode, the server control displays the latest posts. The default is Active.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(eDisplayMode.Active)]
[Description("Choose between Active or Recent. In Active mode, the server control displays the
most active posts. In Recent mode, the server control displays the latest posts.")]
public eDisplayMode DisplayMode;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(eDisplayMode.Active)>
<Description("Choose between Active or Recent. In Active mode, the server control displays the
most active posts. In Recent mode, the server control displays the latest posts.")>
Public Property DisplayMode() As eDisplayMode
```

ActiveTopics.DynamicForumParameter Property

Gets or Sets the QueryString parameter to read a Forum (see page 264) ID dynamically.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Forum ID dynamically.")]
[DefaultValue("f")]
public string DynamicForumParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Forum ID dynamically.")>
<DefaultValue("f")>
Public Property DynamicForumParameter() As string
```

ActiveTopics.DynamicGroupParameter Property

Gets or Sets the QueryString parameter to read a Forum (see page 264) ID dynamically.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Group ID dynamically.")]
[DefaultValue("g")]
public string DynamicGroupParameter;
```

Visual Basic

```
<Bindable(true)>
```

```
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Group ID dynamically.")>
<DefaultValue("g")>
Public Property DynamicGroupParameter() As string
```

ActiveTopics.DynamicThreadParameter Property

Gets or Sets the QueryString parameter to read a Forum (see page 264) ID dynamically.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Thread ID dynamically.")]
[DefaultValue("t")]
public string DynamicThreadParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Thread ID dynamically.")>
<DefaultValue("t")>
Public Property DynamicThreadParameter() As string
```

ActiveTopics.MaxNumber Property

The maximum number of topics listed. The default is 10.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of topics to return.")]
public int MaxNumber;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("The maximum number of topics to return.")>
Public Property MaxNumber() As Integer
```

ActiveTopics.URLPath Property

The URL path to the Forum (see page 264) server control's Web page. For example:

`http://[Web Root]/[Site Root]/forum.aspx`

or

`/[Site Root]/forum.aspx`

If your Forum (see page 264) page and your Active Topics page are in the same folder, you can just enter the name of the page. For example:

`forum.aspx`

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("The URL path to the Forum server control?s Web page.")]
public string URLPath;
```

Visual Basic

```
<Bindable(true)>
```

```
<Category("Ektron")>
<DefaultValue("")>
<Description("The URL path to the Forum server control?s Web page.")>
Public Property URLPath() As string
```

ActivityStream Class

Class Hierarchy

C#

```
[DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:ActivityStream
runat=server></{0}:ActivityStream>")]
public class ActivityStream : EkXsltWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:ActivityStream
runat=server></{0}:ActivityStream>")>
Public Class ActivityStream
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

File

ActivityStream.cs

Description

This is class Ektron.Cms.Controls.ActivityStream.

ActivityStream Methods

The methods of the ActivityStream class are listed here.

Public Methods

	Name	Description
	Fill (see page 20)	Populates the data for the Control.
	LoadData (see page 20)	Binds the ActivityStream (see page 20) control to a custom list of ActivityData objects.
	RaiseCallbackEvent (see page 21)	

ActivityStream.Fill Method

Populates the data for the Control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

ActivityStream.LoadData Method

Binds the ActivityStream (see page 20) control to a custom list of ActivityData objects.

C#

```
public void LoadData(List<ActivityData> activityList);
```

Visual Basic

```
Public Function LoadData(activityList As List<ActivityData>) As void
```

ActivityStream.RaiseCallbackEvent Method

C#

```
public void RaiseCallbackEvent(string eventArgs);
```

Visual Basic

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

Parameters

Parameters	Description
string eventArgs	string

ActivityStream Properties

The properties of the ActivityStream class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 21)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DefaultObjectID (see page 22)	The default object ID for this control to use when there is no matching dynamic parameter value passed.
	DisplayXslt (see page 22)	Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.
	DynamicObjectParameter (see page 22)	Gets or sets the Querystring parameter to read a object ID dynamically. To use the default object ID, leave blank.
	EnablePaging (see page 23)	Enables paging if the total records exceeds MaxResult value. This property, in conjunction with the MaxResults (see page 23) property, lets site visitors view an unlimited number of document items while controlling the amount of screen space. To accomplish this, the display is limited to the number set in the MaxResults (see page 23) property.If you set this property to true, and the number of items exceeds the MaxResults (see page 23) number, navigation aids appear below the last item.
	MaxResults (see page 23)	Maximum number of results to be returned.
	ObjectType (see page 24)	The type of object to which the message board is assigned. Choices are: • **User (see page 950)** - the activity feed is generated for an individual. • **CommunityGroup (see page 750)** - the activity feed is generated for a community group.
	ProfileParamName (see page 24)	The name of the parameter to pass to the profile template.
	Stylesheet (see page 24)	Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet.
	TemplateUserProfile (see page 25)	The URL to launch a new window with, when a member clicks the display-name shown in the messages' browse users screen (used to get more information on a potential message recipient). Optional.
	XmlDoc (see page 25)	Returns the Xslt object for the collection.

ActivityStream.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control's data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control's data is cached. For example, if
```

```
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control’s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control’s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

ActivityStream.DefaultObjectID Property

The default object ID for this control to use when there is no matching dynamic parameter value passed.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The default object ID for this control to use when there is no matching dynamic
parameter value passed.")]
public long DefaultObjectID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The default object ID for this control to use when there is no matching dynamic
parameter value passed.")>
Public Property DefaultObjectID() As long
```

ActivityStream.DisplayXslt Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltActivityStreamTypeConverter))]
[DefaultValue("Builtin")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltActivityStreamTypeConverter))>
<DefaultValue("Builtin")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

ActivityStream.DynamicObjectParameter Property

Gets or sets the Querystring parameter to read a object ID dynamically. To use the default object ID, leave blank.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Gets or sets the Querystring parameter to read a object ID dynamically. To use
the default object ID, leave blank.")]
[Personalizable()]
public string DynamicObjectParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Gets or sets the Querystring parameter to read a object ID dynamically. To use
the default object ID, leave blank.")>
<Personalizable()>
Public Property DynamicObjectParameter() As string
```

ActivityStream.EnablePaging Property

Enables paging if the total records exceeds MaxResult value. This property, in conjunction with the MaxResults (see page 23) property, lets site visitors view an unlimited number of document items while controlling the amount of screen space. To accomplish this, the display is limited to the number set in the MaxResults (see page 23) property.If you set this property to true, and the number of items exceeds the MaxResults (see page 23) number, navigation aids appear below the last item.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Enables paging if total records exceeds MaxResult value. This property, in
conjunction with the MaxResults property, lets site visitors view an unlimited number of
document items while controlling the amount of screen space. To accomplish this, the display
is limited to the number set in the MaxResults property. If you set this property to true, and
the number of items exceeds the MaxResults number, navigation aids appear below the last
item.")]
[Personalizable()]
public bool EnablePaging;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Enables paging if total records exceeds MaxResult value. This property, in
conjunction with the MaxResults property, lets site visitors view an unlimited number of
document items while controlling the amount of screen space. To accomplish this, the display
is limited to the number set in the MaxResults property. If you set this property to true, and
the number of items exceeds the MaxResults number, navigation aids appear below the last
item.")>
<Personalizable()>
Public Property EnablePaging() As bool
```

ActivityStream.MaxResults Property

Maximum number of results to be returned.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of results to fetch. If set to zero, there is no limit.")]
```

```
[Personalizable()]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("The maximum number of results to fetch. If set to zero, there is no limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

ActivityStream.ObjectType Property

The type of object to which the message board is assigned. Choices are:

- **User (see page 950)** - the activity feed is generated for an individual.
- **CommunityGroup (see page 750)** - the activity feed is generated for a community group.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Select the type of object with which this control is associated. Choices are:
User and CommunityGroup.")]
[DefaultValue(EkEnumeration.ActivityFeedType.User)]
public EkEnumeration.ActivityFeedType ObjectType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Select the type of object with which this control is associated. Choices are:
User and CommunityGroup.")>
<DefaultValue(EkEnumeration.ActivityFeedType.User)>
Public Property ObjectType() As EkEnumeration.ActivityFeedType
```

ActivityStream.ProfileParamName Property

The name of the parameter to pass to the profile template.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The parameter name to pass on the querystring to the profile and location pages,
which will be set to the ID of the user that was clicked on.")]
[DefaultValue("id")]
public string ProfileParamName;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The parameter name to pass on the querystring to the profile and location pages,
which will be set to the ID of the user that was clicked on.")>
<DefaultValue("id")>
Public Property ProfileParamName() As string
```

ActivityStream.Stylesheet Property

Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
```

```
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control. Select either 'Use Default', 'None', or enter a
path to a custom Css file. The location can be relative or absolute.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control. Select either 'Use Default', 'None', or enter a
path to a custom Css file. The location can be relative or absolute.")>
<Personalizable()>
Public Property Stylesheet() As string
```

ActivityStream.TemplateUserProfile Property

The URL to launch a new window with, when a member clicks the display-name shown in the messages' browse users screen (used to get more information on a potential message recipient). Optional.

C#

```
[Category("Ektron")]
[Description("The URL to launch a new window with, when a member clicks the avatar shown in
the activitystream browse users screen (used to get more information on a potential message
recipient). Optional.")]
[TypeConverter(typeof(string))]
[DefaultValue("")]
public string TemplateUserProfile;
```

Visual Basic

```
<Category("Ektron")>
<Description("The URL to launch a new window with, when a member clicks the avatar shown in
the activitystream browse users screen (used to get more information on a potential message
recipient). Optional.")>
<TypeConverter(typeof(string))>
<DefaultValue("")>
Public Property TemplateUserProfile() As string
```

ActivityStream.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

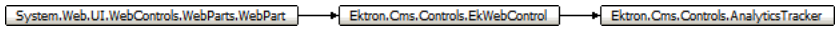
Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

AnalyticsTracker Class

[deprecated]AnalyticsTracker server control has been deprecated. Instead use Analytic server control to track statistics about visits to your Web site.

Class Hierarchy



C#

```
[Obsolete("AnalyticsTracker has been deprecated. Instead control is using Analytics.")]
[DefaultProperty(""), ToolboxData("<{0}:AnalyticsTracker
runat=server></{0}:AnalyticsTracker>")]
public class AnalyticsTracker : EkWebControl;
```

Visual Basic

```
<Obsolete("AnalyticsTracker has been deprecated. Instead control is using Analytics.")>
<DefaultProperty(""), ToolboxData("<{0}:AnalyticsTracker
runat=server></{0}:AnalyticsTracker>")>
Public Class AnalyticsTracker
Inherits EkWebControl
```

File

AnalyticsTracker.cs

AnalyticsTracker Methods

The methods of the AnalyticsTracker class are listed here.

Public Methods

	Name	Description
	Wrap (see page 26)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

AnalyticsTracker.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

AnalyticsTracker Properties

The properties of the AnalyticsTracker class are listed here.

Public Properties

	Name	Description
	DefaultContentID (see page 26)	The Content (see page 822) ID to flag as "visited" in the event that no other ID is specified by QueryString (e.g. "?id=2"). Set to zero (0) if you do not wish to use a Default Content (see page 822) ID.
	DynamicParameter (see page 27)	Gets or Sets the QueryString parameter to read a Content (see page 822) ID dynamically. Set to "" (empty) to always display the default ContentID (static.)
	EnableAnalytics (see page 27)	Determines if the Analytics (see page 745) feature is enabled for this page and (if identified) the content item. Choices are: • **ConfigSpecified** - Take the value from the web.config file's enableAnalytics property. • **True** - Enable Analytic tracking • **False** - Disable Analytic tracking

AnalyticsTracker.DefaultContentID Property

The Content (see page 822) ID to flag as "visited" in the event that no other ID is specified by QueryString (e.g. "?id=2"). Set to

zero (0) if you do not wish to use a Default Content (see page 822) ID.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(AnalyticsTrackerUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("The Content ID to flag as \"visited\" in the event that no other ID is specified
by QueryString (e.g. \"?id=2\"). Set to zero (0) if you do not wish to use a Default Content
ID.")]
public Int64 DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(AnalyticsTrackerUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("The Content ID to flag as \"visited\" in the event that no other ID is specified
by QueryString (e.g. \"?id=2\"). Set to zero (0) if you do not wish to use a Default Content
ID.")>
Public Property DefaultContentID() As Int64
```

AnalyticsTracker.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 822) ID dynamically. Set to "" (empty) to always display the default ContentID (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Content ID dynamically - Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("id")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Content ID dynamically - Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("id")>
Public Property DynamicParameter() As string
```

AnalyticsTracker.EnableAnalytics Property

Determines if the Analytics (see page 745) feature is enabled for this page and (if identified) the content item. Choices are:

- **ConfigSpecified** - Take the value from the web.config file's enableAnalytics property.
- **True** - Enable Analytic tracking
- **False** - Disable Analytic tracking

C#

```
public ConfigOrBool EnableAnalytics;
```

Visual Basic

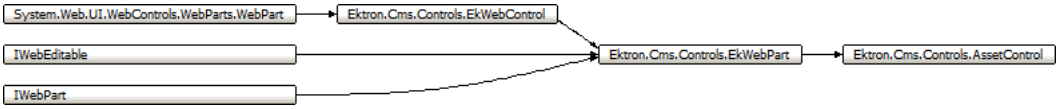
```
Public Property EnableAnalytics() As ConfigOrBool
```

AssetControl Class

The AssetControl server control, when viewed on a Web form, displays a drag and drop box for users to upload assets to

CMS400.NET or update an existing asset.

Class Hierarchy



C#

```
public class AssetControl : EkWebPart;
```

Visual Basic

```
Public Class AssetControl
Inherits EkWebPart
```

File

DocumentControl.cs

AssetControl Methods

The methods of the AssetControl class are listed here.

Public Methods

	Name	Description
	Fill (see page 28)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 28)	Overrides the system ToString function.

AssetControl.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

AssetControl.ToString Method

Overrides the system ToString function.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Returns

String

AssetControl Properties

The properties of the AssetControl class are listed here.

Public Properties

	Name	Description
	DefaultAssetID (see page 29)	The ID of the asset you want to update. This property is used when the UploadType (see page 31) property is set to Update.

	DefaultFolderID (see page 29)	The ID of the folder where assets are added. This property is used when the UploadType (see page 31) property is set to Add. Root is 0.
	DynamicParameter (see page 30)	Gets or Sets the QueryString parameter to read a Content (see page 822) ID or Folder (see page 683) ID dynamically. Content (see page 822) ID will be read if mode is Update. Folder (see page 683) ID will be read if mode is Add. Set to String.Empty if you wish to always display the default DocumentID or FolderID (static.)
	IsImage (see page 30)	Setting this control to 1 (one) restricts the control so only images can be uploaded. • **1** (one) - restrict the control to uploading images only. • **0** (zero) - upload all types of assets.
	OverrideExtension (see page 30)	Allows you to restrict the type of asset that can be uploaded by its extension. For example, to restrict the control to uploading Word documents, enter doc in the property. **Important:** When using this property, enter only the extension's letters not the wildcard (*) or the dot (.). You can add multiple extensions by creating a comma separated list of extensions. Ektron recommends limiting the list to five extensions.
	TaxonomyID (see page 31)	The ID of the taxonomy where assets are added.
	UploadType (see page 31)	Selects whether the control will add a document to a folder or overwrite an existing document. Select **Add** to add assets and use the DefaultFolderID (see page 29) property. Select **Update** to update assets.

AssetControl.DefaultAssetID Property

The ID of the asset you want to update. This property is used when the UploadType (see page 31) property is set to Update.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("The ID of the asset you want to update. This property is used when the
UploadType property is set to Update.")]
[Personalizable(), WebBrowsable()]
public long DefaultAssetID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("The ID of the asset you want to update. This property is used when the
UploadType property is set to Update.")>
<Personalizable(), WebBrowsable()>
Public Property DefaultAssetID() As long
```

AssetControl.DefaultFolderID Property

The ID of the folder where assets are added. This property is used when the UploadType (see page 31) property is set to Add. Root is 0.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The ID of the folder where assets are added. This property is used when the
UploadType property is set to Add. Root is 0.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long DefaultFolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
```

```
<Browsable(true)>
<Description("The ID of the folder where assets are added. This property is used when the
UploadType property is set to Add. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property DefaultFolderID() As long
```

AssetControl.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 822) ID or Folder (see page 683) ID dynamically. Content (see page 822) ID will be read if mode is Update. Folder (see page 683) ID will be read if mode is Add. Set to String.Empty if you wish to always display the default DocumentID or FolderID (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Content ID or Folder ID
dynamically. Content ID will be read if mode is Update. Folder ID will be read if mode is Add.
Set to String.Empty if you wish to always display the default DocumentID or FolderID
(static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
[Personalizable(), WebBrowsable()]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Content ID or Folder ID
dynamically. Content ID will be read if mode is Update. Folder ID will be read if mode is Add.
Set to String.Empty if you wish to always display the default DocumentID or FolderID
(static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
<Personalizable(), WebBrowsable()>
Public Property DynamicParameter() As string
```

AssetControl.IsImage Property

Setting this control to 1 (one) restricts the control so only images can be uploaded.

- **1** (one) - restrict the control to uploading images only.
- **0** (zero) - upload all types of assets.

C#

```
[Category("Ektron")]
[DefaultValue("0")]
[Description("Setting this control to 1 (one) restricts the control so only images can be
uploaded. 1 - restrict the control to uploading images only. 0 - upload all types of assets.")]
public int IsImage;
```

Visual Basic

```
<Category("Ektron")>
<DefaultValue("0")>
<Description("Setting this control to 1 (one) restricts the control so only images can be
uploaded. 1 - restrict the control to uploading images only. 0 - upload all types of assets.")>
Public Property IsImage() As Integer
```

AssetControl.OverrideExtension Property

Allows you to restrict the type of asset that can be uploaded by its extension. For example, to restrict the control to uploading Word documents, enter doc in the property.

Important: When using this property, enter only the extension's letters not the wildcard (*) or the dot (.).

You can add multiple extensions by creating a comma separated list of extensions. Ektron recommends limiting the list to five extensions.

C#

```
[Category("Ektron")]
[DefaultValue("0")]
[Description("Allows you to restrict the type of asset that can be uploaded by its extension.
For example, to restrict the control to uploading Word documents, enter doc in the property.
Important: When using this property, enter only the extension?s letters not the wildcard (*)
or the dot (.).")]
public string OverrideExtension;
```

Visual Basic

```
<Category("Ektron")>
<DefaultValue("0")>
<Description("Allows you to restrict the type of asset that can be uploaded by its extension.
For example, to restrict the control to uploading Word documents, enter doc in the property.
Important: When using this property, enter only the extension?s letters not the wildcard (*)
or the dot (.).")>
Public Property OverrideExtension() As string
```

AssetControl.TaxonomyID Property

The ID of the taxonomy where assets are added.

C#

```
[Category("Ektron")]
[DefaultValue("0")]
[Description("The ID of the taxonomy where assets are added.")]
[Personalizable(), WebBrowsable()]
public long TaxonomyID;
```

Visual Basic

```
<Category("Ektron")>
<DefaultValue("0")>
<Description("The ID of the taxonomy where assets are added.")>
<Personalizable(), WebBrowsable()>
Public Property TaxonomyID() As long
```

AssetControl.UploadType Property

Selects whether the control will add a document to a folder or overwrite an existing document. Select **Add** to add assets and use the DefaultFolderID (see page 29) property. Select **Update** to update assets.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Selects whether the control will add a document to a folder or overwrite an
existing document.")]
[Personalizable(), WebBrowsable()]
public UploadTypeEnum UploadType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Selects whether the control will add a document to a folder or overwrite an
existing document.")>
<Personalizable(), WebBrowsable()>
Public Property UploadType() As UploadTypeEnum
```

Blog Class

The Blog server control allows you to quickly add a blog to a Web form. It has all of the items commonly found on a blog page in one server control.

Class Hierarchy

C#

```
[DefaultProperty("BlogID"), ToolboxData("<{0}:Blog runat=server></{0}:Blog>")]
public class Blog : EkWebPart;
```

Visual Basic

```
<DefaultProperty("BlogID"), ToolboxData("<{0}:Blog runat=server></{0}:Blog>")>
Public Class Blog
Inherits EkWebPart
```

File

Blog.cs

Blog Methods

The methods of the Blog class are listed here.

Public Methods

	Name	Description
	Fill (see page 32)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 32)	Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.
	Wrap (see page 33)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Blog.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Blog.ToString Method

Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Blog.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

Blog Properties

The properties of the Blog class are listed here.

Public Properties

	Name	Description
	ArchiveMode (see page 33)	Select whether the archive appears in month format or year format. The default is month.
	BlogID (see page 34)	The ID of the blog to display when the blog is not associated with a specific user. To display a blog associated with a user, leave this property set to 0 (zero) and enter the user's ID in the DefaultUserID (see page 35) property.
	CacheInterval (see page 34)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300.
	DateToStart (see page 35)	The date of the last blog posts you want to appear. For example, if you want to display blog posts for January 1, 2007 and before, you enter 1/1/2007.
	DefaultUserID (see page 35)	The ID of the user who owns the blog to be displayed. This is used when the server control displays a user's blog. To display a blog not associated with a user, leave this property set to 0 (zero) and enter the blog's ID in the BlogID (see page 34) property.
	DynamicParameter (see page 35)	Gets or Sets the QueryString parameter to read a Blog (see page 32) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 32) (static.)
	DynamicUserParameter (see page 36)	Gets or sets the QueryString parameter to read a User (see page 950) ID dynamically. Set to String.Empty if you wish to always display the default user's blog (static.)
	EditorHeight (see page 36)	Sets the height of the blog editor in pixels.
	EditorWidth (see page 36)	Sets the width of the blog editor in pixels.
	JavascriptEditorHTMLMode (see page 37)	Set to True to allow users to edit their posts and replies in HTML.
	MaxResults (see page 37)	Set the maximum number of posts to display. The default is -3. • If set to 0 (zero), there is no limit. • If set to -1, all posts for the day are shown. • If set to -2, all post for the month are shown. • If set to -3, the control uses the # of Post Visible setting in the Workarea.
	PostParameter (see page 37)	Checks the QueryString for this value and shows the post when specified. Leave blank to always display the blog.
	RecentPosts (see page 38)	Sets the number of recent posts to display. Integer value, must be greater than 0 (Zero).
	ShowHeader (see page 38)	Shows the title and tagline when set to True.
	ShowRSS (see page 38)	Displays the icon for the RSS feed when set to True.

Blog.ArchiveMode Property

Select whether the archive appears in month format or year format. The default is month.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(EkEnumeration.BlogArchiveMode.Month)]
[Description("Select whether the archive appears in month format or year format. The default
is month.")]
[Localizable(true)]
public EkEnumeration.BlogArchiveMode ArchiveMode;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(EkEnumeration.BlogArchiveMode.Month)>
<Description("Select whether the archive appears in month format or year format. The default
is month.")>
<Localizable(true)>
Public Property ArchiveMode() As EkEnumeration.BlogArchiveMode
```

Blog.BlogID Property

The ID of the blog to display when the blog is not associated with a specific user. To display a blog associated with a user, leave this property set to 0 (zero) and enter the user's ID in the DefaultUserID (see page 35) property.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the blog to display when the blog is not associated with a specific
user. To display a blog associated with a user, leave this property set to 0 (zero) and enter
the user's ID in the DefaultUserID property.")]
[Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long BlogID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the blog to display when the blog is not associated with a specific
user. To display a blog associated with a user, leave this property set to 0 (zero) and enter
the user's ID in the DefaultUserID property.")>
<Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property BlogID() As long
```

Blog.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300.")]
[DefaultValue(0)]
public double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
```

```
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300.")>
<DefaultValue(0)>
Public Property CacheInterval() As double
```

Blog.DateToStart Property

The date of the last blog posts you want to appear. For example, if you want to display blog posts for January 1, 2007 and before, you enter 1/1/2007.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Date from which to get posts.")]
public DateTime DateToStart;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Date from which to get posts.")>
Public Property DateToStart() As DateTime
```

Blog.DefaultUserID Property

The ID of the user who owns the blog to be displayed. This is used when the server control displays a user's blog. To display a blog not associated with a user, leave this property set to 0 (zero) and enter the blog's ID in the BlogID (see page 34) property.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the user who owns the blog to be displayed. This is used when the
server control displays a user's blog. To display a blog not associated with a user, leave
this property set to 0 (zero) and enter the blog's ID in the BlogID property.")]
[Personalizable(), WebBrowsable()]
public long DefaultUserID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the user who owns the blog to be displayed. This is used when the
server control displays a user's blog. To display a blog not associated with a user, leave
this property set to 0 (zero) and enter the blog's ID in the BlogID property.")>
<Personalizable(), WebBrowsable()>
Public Property DefaultUserID() As long
```

Blog.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Blog (see page 32) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 32) (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or sets the QueryString parameter to read a Blog ID dynamically. Set to
\"Use Default\" if you wish to always display the default blog (static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("blogid")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or sets the QueryString parameter to read a Blog ID dynamically. Set to
\"Use Default\" if you wish to always display the default blog (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("blogid")>
Public Property DynamicParameter() As string
```

Blog.DynamicUserParameter Property

Gets or sets the QueryString parameter to read a User (see page 950) ID dynamically. Set to String.Empty if you wish to always display the default user's blog (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or sets the QueryString parameter to read a User ID dynamically. Set to
\"Use Default\" if you wish to always display the default user's blog (static.)")]
[TypeConverter(typeof(DynamicUserIDTypeConverter))]
[DefaultValue("")]
public string DynamicUserParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or sets the QueryString parameter to read a User ID dynamically. Set to
\"Use Default\" if you wish to always display the default user's blog (static.)")>
<TypeConverter(typeof(DynamicUserIDTypeConverter))>
<DefaultValue("")>
Public Property DynamicUserParameter() As string
```

Blog.EditorHeight Property

Sets the height of the blog editor in pixels.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(400)]
[Description("The pixel height for the Editor.")]
public int EditorHeight;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(400)>
<Description("The pixel height for the Editor.")>
Public Property EditorHeight() As Integer
```

Blog.EditorWidth Property

Sets the width of the blog editor in pixels.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(625)]
[Description("The pixel width for the Editor.")]
public int EditorWidth;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(625)>
<Description("The pixel width for the Editor.")>
Public Property EditorWidth() As Integer
```

Blog.JavascriptEditorHTMLMode Property

Set to True to allow users to edit their posts and replies in HTML.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Determines if HTML mode is allowed for the Javascript Editor. True enables HTML
mode.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
public bool JavascriptEditorHTMLMode;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Determines if HTML mode is allowed for the Javascript Editor. True enables HTML
mode.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorHTMLMode() As bool
```

Blog.MaxResults Property

Set the maximum number of posts to display. The default is -3.

- If set to 0 (zero), there is no limit.
- If set to -1, all posts for the day are shown.
- If set to -2, all post for the month are shown.
- If set to -3, the control uses the # of Post Visible setting in the Workarea.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(-3)]
[Description("The maximum number of posts to show. If set to zero, there is no limit. If -1,
all posts for the day are shown. If -2, all post for the month are shown. If set to -3, the
control uses the # of Post Visible setting in the Workarea.")]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(-3)>
<Description("The maximum number of posts to show. If set to zero, there is no limit. If -1,
all posts for the day are shown. If -2, all post for the month are shown. If set to -3, the
control uses the # of Post Visible setting in the Workarea.")>
Public Property MaxResults() As Integer
```

Blog.PostParameter Property

Checks the QueryString for this value and shows the post when specified. Leave blank to always display the blog.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("id")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and replaces the blog with that post when
specified.")]
public string PostParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("id")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and replaces the blog with that post when
specified.")>
Public Property PostParameter() As string
```

Blog.RecentPosts Property

Sets the number of recent posts to display. Integer value, must be greater than 0 (Zero).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Number of Recent Posts to Display. Value must be greater than 0 (Zero).")]
[DefaultValue(5)]
public int RecentPosts;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Number of Recent Posts to Display. Value must be greater than 0 (Zero).")>
<DefaultValue(5)>
Public Property RecentPosts() As Integer
```

Blog.ShowHeader Property

Shows the title and tagline when set to True.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Shows the title and tagline when set to True.")]
public bool ShowHeader;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Shows the title and tagline when set to True.")>
Public Property ShowHeader() As bool
```

Blog.ShowRSS Property

Displays the icon for the RSS feed when set to True.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
```

```
[Localizable(true)]
[Description("Displays the icon for the RSS feed when set to True.")]
public bool ShowRSS;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Localizable(true)>
<Description("Displays the icon for the RSS feed when set to True.")>
Public Property ShowRSS() As bool
```

BlogArchive Class

The BlogArchive server control displays a list of past months or years that have posts.

Class Hierarchy

C#

```
[DefaultProperty("BlogID")]
[ToolboxData("<{0}:BlogArchive runat=server></{0}:BlogArchive>")]
public class BlogArchive : EkWebControl;
```

Visual Basic

```
<DefaultProperty("BlogID")>
<ToolboxData("<{0}:BlogArchive runat=server></{0}:BlogArchive>")>
Public Class BlogArchive
Inherits EkWebControl
```

File

BlogArchive.cs

BlogArchive Methods

The methods of the BlogArchive class are listed here.

Public Methods

	Name	Description
	Fill (see page 39)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 39)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

BlogArchive.Fill Method

Overrideable method that can be used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

BlogArchive.Wrap Method

Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

BlogArchive Properties

The properties of the BlogArchive class are listed here.

Public Properties

	Name	Description
	ArchiveMode (see page 40)	Select whether the archive appears in month format or year format. The default is month.
	BlogID (see page 40)	The ID of the blog in CMS400.NET. Can be selected via the CMS Explorer in design time.
	CacheInterval (see page 41)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DynamicParameter (see page 41)	Gets or Sets the QueryString parameter to read a Blog (see page 32) ID dynamically. Set to "" if you wish to always display the default blog.
	Text (see page 41)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.
	WrapTag (see page 42)	Allows a developer to specify control's tag. Default is assumed "".

BlogArchive.ArchiveMode Property

Select whether the archive appears in month format or year format. The default is month.

C#

```
public Common.EkEnumeration.BlogArchiveMode ArchiveMode;
```

Visual Basic

```
Public Property ArchiveMode() As Common.EkEnumeration.BlogArchiveMode
```

BlogArchive.BlogID Property

The ID of the blog in CMS400.NET. Can be selected via the CMS Explorer in design time.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the blog for which to display an archive.")]
[Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BlogID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the blog for which to display an archive.")>
<Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BlogID() As long
```

BlogArchive.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
Public virtual Property CacheInterval() As double
```

BlogArchive.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Blog (see page 32) ID dynamically. Set to "" if you wish to always display the default blog.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or sets the QueryString parameter to read a Blog ID dynamically. Set to
\"\" if you wish to always display the default blog.")]
[TypeConverter(typeof(DynamicBlogIDTypeConverter))]
[DefaultValue("blogid")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or sets the QueryString parameter to read a Blog ID dynamically. Set to
\"\" if you wish to always display the default blog.")>
<TypeConverter(typeof(DynamicBlogIDTypeConverter))>
<DefaultValue("blogid")>
Public Property DynamicParameter() As string
```

BlogArchive.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

```
public override string Text;
```

Visual Basic

```
Public override Property Text() As string
```

BlogArchive.WrapTag Property

Allows a developer to specify control's tag. Default is assumed "".

C#

```
[DefaultValue("div")]
public new string WrapTag;
```

Visual Basic

```
<DefaultValue("div")>
Public new Property WrapTag() As string
```

BlogCalendar Class

The BlogCalendar server control allows a user to display a calendar on a Web page and associate it with a blog.

Class Hierarchy

C#

```
[DefaultProperty("BlogID")]
[ToolboxData("<{0}:BlogCalendar runat=server></{0}:BlogCalendar>")]
public class BlogCalendar : EkWebControl;
```

Visual Basic

```
<DefaultProperty("BlogID")>
<ToolboxData("<{0}:BlogCalendar runat=server></{0}:BlogCalendar>")>
Public Class BlogCalendar
Inherits EkWebControl
```

File

BlogCalendar.cs

BlogCalendar Methods

The methods of the BlogCalendar class are listed here.

Public Methods

	Name	Description
	Fill (see page 42)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 42)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

BlogCalendar.Fill Method

Overrideable method that can be used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

BlogCalendar.Wrap Method

Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

Html string wrapped with a toolbar.

BlogCalendar Properties

The properties of the BlogCalendar class are listed here.

Public Properties

	Name	Description
	BlogID (see page 43)	The ID of the blog for which to display a calendar. Can be selected via the CMS Explorer in design time.
	CacheInterval (see page 43)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	Text (see page 44)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.
	WrapTag (see page 44)	Allows a developer to specify control's tag. Default is assumed " ".

BlogCalendar.BlogID Property

The ID of the blog for which to display a calendar. Can be selected via the CMS Explorer in design time.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the blog for which to display a calendar for.")]
[Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BlogID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the blog for which to display a calendar for.")>
<Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BlogID() As long
```

BlogCalendar.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Interval to cache this control in second(s).")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Interval to cache this control in second(s).")>
<DefaultValue(0)>
Public virtual Property CacheInterval() As double
```

BlogCalendar.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

```
public override string Text;
```

Visual Basic

```
Public override Property Text() As string
```

BlogCalendar.WrapTag Property

Allows a developer to specify control's tag. Default is assumed " ".

C#

```
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("div")]
[TypeConverter(typeof(WrapTagTypeConverter))]
[Description("Allows a developer to specify control's tag. Default is assumed '<div>'.")]
public new string WrapTag;
```

Visual Basic

```
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("div")>
<TypeConverter(typeof(WrapTagTypeConverter))>
<Description("Allows a developer to specify control's tag. Default is assumed '<div>'.")>
Public new Property WrapTag() As string
```

BlogCategories Class

The BlogCategories server control displays a blog's categories on a Web form.

Class Hierarchy

C#

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:BlogCategories runat=server></{0}:BlogCategories>")]
public class BlogCategories : EkWebControl;
```

Visual Basic

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:BlogCategories runat=server></{0}:BlogCategories>")>
Public Class BlogCategories
Inherits EkWebControl
```

File

BlogCategories.cs

BlogCategories Methods

The methods of the BlogCategories class are listed here.

Public Methods

	Name	Description
	Fill (see page 45)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 45)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

BlogCategories.Fill Method

Overrideable method that can be used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

BlogCategories.Wrap Method

Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

BlogCategories Properties

The properties of the BlogCategories class are listed here.

Public Properties

	Name	Description
	BlogID (see page 45)	The ID of the blog from which to display a list of categories.
	CacheInterval (see page 46)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	Text (see page 46)	Allows a developer to treat the server control as a label, taking information from the Control and painting it on top of the control for display. Leave blank for default display.
	WrapTag (see page 46)	Allows a developer to specify control's tag. Default is assumed " ".

BlogCategories.BlogID Property

The ID of the blog from which to display a list of categories.

C#

```
[Bindable(true)]
[Category("Ektron")]
```

```
[DefaultValue(0)]
[Description("The ID of the blog from which to display a list of categories.")]
[Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BlogID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the blog from which to display a list of categories.")>
<Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BlogID() As long
```

BlogCategories.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
Public virtual Property CacheInterval() As double
```

BlogCategories.Text Property

Allows a developer to treat the server control as a label, taking information from the Control and painting it on top of the control for display. Leave blank for default display.

C#

```
public override string Text;
```

Visual Basic

```
Public override Property Text() As string
```

BlogCategories.WrapTag Property

Allows a developer to specify control's tag. Default is assumed " ".

C#

```
[Browsable(false)]
[DefaultValue("div")]
public new string WrapTag;
```

Visual Basic

```
<Browsable(false)>
<DefaultValue("div")>
```

```
Public new Property WrapTag() As string
```

BlogEntries Class

The BlogEntries server control allows you to display a list of multiple blog posts on a Web form.

Class Hierarchy

C#

```
[DefaultProperty("BlogID"), ToolboxData("<{0}:BlogEntries runat=server></{0}:BlogEntries>")]
public class BlogEntries : EkXsltBase, IEnumerator, IEnumerable, IListSource, IDataSource;
```

Visual Basic

```
<DefaultProperty("BlogID"), ToolboxData("<{0}:BlogEntries runat=server></{0}:BlogEntries>")>
Public Class BlogEntries
Inherits EkXsltBase
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
```

File

BlogEntries.cs

BlogEntries Methods

The methods of the BlogEntries class are listed here.

Public Methods

	Name	Description
	Fill (see page 47)	Method that can be overridden and used to grab data from the back end for each server control.
	GetRssFeed (see page 47)	This method returns a blog post RSS feed.
	GetView (see page 48)	Gets the view data using dataset
	GetViewNames (see page 48)	gets the view names
	ToString (see page 48)	Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.
	Wrap (see page 48)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

BlogEntries.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

BlogEntries.GetRssFeed Method

This method returns a blog post RSS feed.

C#

```
public string GetRssFeed();
```

Visual Basic

```
Public Function GetRssFeed() As string
```

Returns

String

BlogEntries.GetView Method

Gets the view data using dataset

C#

```
public DataSourceView GetView(string viewName);
```

Visual Basic

```
Public Function GetView(viewName As string) As DataSourceView
```

Parameters

Parameters	Description
string viewName	

Returns

DataSourceView

BlogEntries.GetViewNames Method

gets the view names

C#

```
public ICollection GetViewNames();
```

Visual Basic

```
Public Function GetViewNames() As ICollection
```

Returns

View names

BlogEntries.ToString Method

Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

BlogEntries.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

BlogEntries Properties

The properties of the BlogEntries class are listed here.

Public Properties

	Name	Description
	BlogID (see page 49)	The ID of the blog to display when the blog is not associated with a specific user. To display a blog associated with a user, leave this property set to 0 (zero) and enter the user's ID in the DefaultUserID (see page 51) property.
	BlogPostParameter (see page 50)	Checks the QueryString for this value and shows the post when specified. Leave blank to always display the blog.
	CacheInterval (see page 50)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300.
	DateToStart (see page 50)	The date of the last blog posts you want to appear. For example, if you want to display blog posts for January 1, 2007 and before, you enter 1/1/2007.
	DefaultUserID (see page 51)	The ID of the user who owns the blog to be displayed. This is used when the server control displays a user's blog. To display a blog not associated with a user, leave this property set to 0 (zero) and enter the blog's ID in the BlogID (see page 49) property.
	DisplayXslt (see page 51)	The Xslt used to render the blog entry list (optional).
	DynamicParameter (see page 51)	Gets or Sets the QueryString parameter to read a Blog (see page 32) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 32) (static.)
	DynamicUserParameter (see page 52)	Gets or sets the QueryString parameter to read a User (see page 950) ID dynamically. Set to String.Empty if you wish to always display the default user's blog (static.)
	EditorHeight (see page 52)	Sets the height of the blog editor in pixels.
	EditorWidth (see page 52)	Sets the width of the blog editor in pixels.
	JavascriptEditorHTMLMode (see page 53)	Set to True to allow users to edit their posts and replies in HTML.
	MaxResults (see page 53)	Set the maximum number of posts to display. The default is -3. • If set to 0 (zero), there is no limit. • If set to -1, all posts for the day are shown. • If set to -2, all post for the month are shown. • If set to -3, the control uses the # of Post Visible setting in the Workarea.
	ShowHeader (see page 54)	Shows the title and tagline when set to True.
	XmlDoc (see page 54)	Returns the Xslt object for the collection.

BlogEntries.BlogID Property

The ID of the blog to display when the blog is not associated with a specific user. To display a blog associated with a user, leave this property set to 0 (zero) and enter the user's ID in the DefaultUserID (see page 51) property.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the blog to display when the blog is not associated with a specific
user. To display a blog associated with a user, leave this property set to 0 (zero) and enter
```

```
the user's ID in the DefaultUserID property.")]
[Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BlogID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the blog to display when the blog is not associated with a specific
user. To display a blog associated with a user, leave this property set to 0 (zero) and enter
the user's ID in the DefaultUserID property.")>
<Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BlogID() As long
```

BlogEntries.BlogPostParameter Property

Checks the QueryString for this value and shows the post when specified. Leave blank to always display the blog.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("id")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and replaces the blog with that post when
specified.")]
public string BlogPostParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("id")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and replaces the blog with that post when
specified.")>
Public Property BlogPostParameter() As string
```

BlogEntries.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300.")]
[DefaultValue(0)]
public double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300.")>
<DefaultValue(0)>
Public Property CacheInterval() As double
```

BlogEntries.DateToStart Property

The date of the last blog posts you want to appear. For example, if you want to display blog posts for January 1, 2007 and before, you enter 1/1/2007.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Date from which to get posts.")]
public DateTime DateToStart;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Date from which to get posts.")>
Public Property DateToStart() As DateTime
```

BlogEntries.DefaultUserID Property

The ID of the user who owns the blog to be displayed. This is used when the server control displays a user's blog. To display a blog not associated with a user, leave this property set to 0 (zero) and enter the blog's ID in the BlogID (see page 49) property.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the user who owns the blog to be displayed. This is used when the
server control displays a user's blog. To display a blog not associated with a user, leave
this property set to 0 (zero) and enter the blog's ID in the BlogID property.")]
public long DefaultUserID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the user who owns the blog to be displayed. This is used when the
server control displays a user's blog. To display a blog not associated with a user, leave
this property set to 0 (zero) and enter the blog's ID in the BlogID property.")>
Public Property DefaultUserID() As long
```

BlogEntries.DisplayXslt Property

The Xslt used to render the blog entry list (optional).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the Control (optional).")]
[TypeConverter(typeof(XsltBlogTypeConverter))]
[DefaultValue("ecmBlog")]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the Control (optional).")>
<TypeConverter(typeof(XsltBlogTypeConverter))>
<DefaultValue("ecmBlog")>
Public override Property DisplayXslt() As string
```

BlogEntries.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Blog (see page 32) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 32) (static.)

C#

```
[Bindable(true)]
```

```
[Category("Ektron")]
[Description("Gets or sets the QueryString parameter to read a Blog ID dynamically. Set to
\"Use Default\" if you wish to always display the default blog (static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("blogid")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or sets the QueryString parameter to read a Blog ID dynamically. Set to
\"Use Default\" if you wish to always display the default blog (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("blogid")>
Public Property DynamicParameter() As string
```

BlogEntries.DynamicUserParameter Property

Gets or sets the QueryString parameter to read a User (see page 950) ID dynamically. Set to String.Empty if you wish to always display the default user's blog (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or sets the QueryString parameter to read a User ID dynamically. Set to
\"Use Default\" if you wish to always display the default user's blog (static.)")]
[TypeConverter(typeof(DynamicUserIDTypeConverter))]
[DefaultValue("")]
public string DynamicUserParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or sets the QueryString parameter to read a User ID dynamically. Set to
\"Use Default\" if you wish to always display the default user's blog (static.)")>
<TypeConverter(typeof(DynamicUserIDTypeConverter))>
<DefaultValue("")>
Public Property DynamicUserParameter() As string
```

BlogEntries.EditorHeight Property

Sets the height of the blog editor in pixels.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(400)]
[Description("The pixel height for the Editor.")]
public int EditorHeight;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(400)>
<Description("The pixel height for the Editor.")>
Public Property EditorHeight() As Integer
```

BlogEntries.EditorWidth Property

Sets the width of the blog editor in pixels.

C#

```
[Bindable(true)]
```

```
[Category("Ektron")]
[DefaultValue(625)]
[Description("The pixel width for the Editor.")]
public int EditorWidth;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(625)>
<Description("The pixel width for the Editor.")>
Public Property EditorWidth() As Integer
```

BlogEntries.JavascriptEditorHTMLMode Property

Set to True to allow users to edit their posts and replies in HTML.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Determines if HTML mode is allowed for the Javascript Editor. True enables HTML
mode.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
public bool JavascriptEditorHTMLMode;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Determines if HTML mode is allowed for the Javascript Editor. True enables HTML
mode.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorHTMLMode() As bool
```

BlogEntries.MaxResults Property

Set the maximum number of posts to display. The default is -3.

- If set to 0 (zero), there is no limit.
- If set to -1, all posts for the day are shown.
- If set to -2, all post for the month are shown.
- If set to -3, the control uses the # of Post Visible setting in the Workarea.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(-3)]
[Description("The maximum number of posts to show. If set to zero, there is no limit. If -1,
all posts for the day are shown. If -2, all post for the month are shown. If set to -3, the
control uses the # of Post Visible setting in the Workarea.")]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(-3)>
<Description("The maximum number of posts to show. If set to zero, there is no limit. If -1,
all posts for the day are shown. If -2, all post for the month are shown. If set to -3, the
control uses the # of Post Visible setting in the Workarea.")>
Public Property MaxResults() As Integer
```

BlogEntries.ShowHeader Property

Shows the title and tagline when set to True.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Shows the title and tagline when set to True.")]
public bool ShowHeader;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Shows the title and tagline when set to True.")>
Public Property ShowHeader() As bool
```

BlogEntries.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

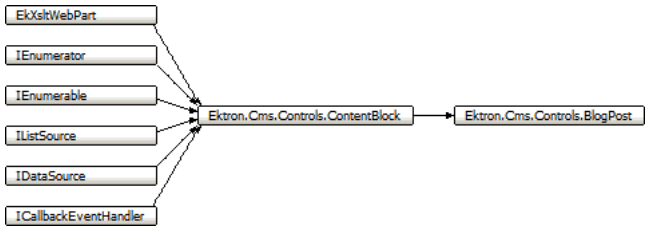
Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

BlogPost Class

The BlogPost server control allows you to display an individual blog post on a page.

Class Hierarchy



C#

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:BlogPost runat=server></{0}:BlogPost>")]
public class BlogPost : ContentBlock;
```

Visual Basic

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:BlogPost runat=server></{0}:BlogPost>")>
Public Class BlogPost
Inherits ContentBlock
```

File

BlogPost.cs

BlogPost Methods

The methods of the BlogPost class are listed here.

Public Methods

	Name	Description
	Wrap (see page 55)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

BlogPost.Wrap Method

Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

BlogPost Properties

The properties of the BlogPost class are listed here.

Public Properties

	Name	Description
	DefaultContentID (see page 55)	The ID of a default blog post that appears where you inserted this server control if no other content block is identified, or is not available.
	ShowType (see page 56)	Shows a blog post's content and its comments or just the blog post's comments. The default is Content (see page 822). • **Content (see page 822)** - Show a blog post's content and its comments. • **Description** - Show a blog post's comments only.
	XmlDoc (see page 56)	Returns the Xslt object for the collection.

BlogPost.DefaultContentID Property

The ID of a default blog post that appears where you inserted this server control if no other content block is identified, or is not available.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(BlogPostUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("The ID of a default blog post that appears where you inserted this server
control if no other content block is identified, or is not available.")]
public new long DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(BlogPostUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("The ID of a default blog post that appears where you inserted this server
control if no other content block is identified, or is not available.")>
```

```
Public new Property DefaultContentID() As long
```

BlogPost.ShowType Property

Shows a blog post's content and its comments or just the blog post's comments. The default is Content (see page 822).

- **Content (see page 822)** - Show a blog post's content and its comments.
- **Description** - Show a blog post's comments only.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Content")]
[Description("Shows a blog post?s content and its comments or just the blog post?s comments.
The default is Content. \"Content\" shows a blog post?s content and its comments.
\"Description\" shows a blog post?s comments only.")]
[Localizable(true)]
public eShowType ShowType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Content")>
<Description("Shows a blog post?s content and its comments or just the blog post?s comments.
The default is Content. \"Content\" shows a blog post?s content and its comments.
\"Description\" shows a blog post?s comments only.")>
<Localizable(true)>
Public Property ShowType() As eShowType
```

BlogPost.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

BlogRecentPosts Class

The BlogRecentPosts server control displays a list of recent blog posts links on a Web form.

Class Hierarchy

C#

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:BlogRecentPosts runat=server></{0}:BlogRecentPosts>")]
public class BlogRecentPosts : EkWebControl;
```

Visual Basic

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:BlogRecentPosts runat=server></{0}:BlogRecentPosts>")>
Public Class BlogRecentPosts
Inherits EkWebControl
```

File

BlogRecentPosts.cs

BlogRecentPosts Methods

The methods of the BlogRecentPosts class are listed here.

Public Methods

	Name	Description
	Fill (see page 57)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 57)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

BlogRecentPosts.Fill Method

Overrideable method that can be used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

BlogRecentPosts.Wrap Method

Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

BlogRecentPosts Properties

The properties of the BlogRecentPosts class are listed here.

Public Properties

	Name	Description
	BlogID (see page 57)	The ID of the blog from which to select recent posts. Can be selected via the CMS Explorer in design time.
	CacheInterval (see page 58)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache data for five minutes, you set the CacheInterval property to 300 (three hundred).
	NumberofPosts (see page 58)	Sets the number of recent posts to display. Integer value, must be greater than 0 (zero).
	Text (see page 59)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.
	WrapTag (see page 59)	Allows a developer to specify control's tag. Default is assumed " ".

BlogRecentPosts.BlogID Property

The ID of the blog from which to select recent posts. Can be selected via the CMS Explorer in design time.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the blog from which to select recent posts.")]
[Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BlogID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the blog from which to select recent posts.")>
<Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BlogID() As long
```

BlogRecentPosts.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
Public virtual Property CacheInterval() As double
```

BlogRecentPosts.NumberofPosts Property

Sets the number of recent posts to display. Integer value, must be greater than 0 (zero).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the number of recent posts to display.")]
[DefaultValue(5)]
public int NumberofPosts;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the number of recent posts to display.")>
<DefaultValue(5)>
Public Property NumberofPosts() As Integer
```

BlogRecentPosts.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

```
public override string Text;
```

Visual Basic

```
Public override Property Text() As string
```

BlogRecentPosts.WrapTag Property

Allows a developer to specify control's tag. Default is assumed " ".

C#

```
[Browsable(false)]
[DefaultValue("div")]
public new string WrapTag;
```

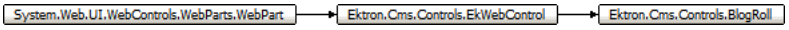
Visual Basic

```
<Browsable(false)>
<DefaultValue("div")>
Public new Property WrapTag() As string
```

BlogRoll Class

The BlogRoll server control displays the blog roll for a blog on a Web form.

Class Hierarchy



C#

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:BlogRoll runat=server></{0}:BlogRoll>")]
public class BlogRoll : EkWebControl;
```

Visual Basic

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:BlogRoll runat=server></{0}:BlogRoll>")>
Public Class BlogRoll
Inherits EkWebControl
```

File

BlogRoll.cs

BlogRoll Methods

The methods of the BlogRoll class are listed here.

Public Methods

	Name	Description
	Fill (see page 59)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 60)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

BlogRoll.Fill Method

Overrideable method that can be used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

BlogRoll.Wrap Method

Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

BlogRoll Properties

The properties of the BlogRoll class are listed here.

Public Properties

	Name	Description
	BlogID (see page 60)	The ID of the blog in CMS400.NET. Can be selected via the CMS Explorer in design time.
	CacheInterval (see page 61)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	Text (see page 61)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.
	WrapTag (see page 61)	Allows a developer to specify control's tag. Default is assumed " ".

BlogRoll.BlogID Property

The ID of the blog in CMS400.NET. Can be selected via the CMS Explorer in design time.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("\\")]
[Description("The ID of the blog from which to select entries.")]
[Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BlogID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("\\")>
<Description("The ID of the blog from which to select entries.")>
<Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BlogID() As long
```

BlogRoll.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
Public virtual Property CacheInterval() As double
```

BlogRoll.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

```
public override string Text;
```

Visual Basic

```
Public override Property Text() As string
```

BlogRoll.WrapTag Property

Allows a developer to specify control's tag. Default is assumed " ".

C#

```
[Browsable(false)]
[DefaultValue("div")]
public new string WrapTag;
```

Visual Basic

```
<Browsable(false)>
<DefaultValue("div")>
Public new Property WrapTag() As string
```

BlogRSS Class

The BlogRSS server control displays the icon for the blog's RSS feed on the Web form.

Class Hierarchy

C#

```
[DefaultProperty("Text")]
```

```
[ToolboxData("<{0}:BlogRSS runat=server></{0}:BlogRSS>")]
public class BlogRSS : EkWebControl;
```

Visual Basic

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:BlogRSS runat=server></{0}:BlogRSS>")>
Public Class BlogRSS
Inherits EkWebControl
```

File

BlogRSS.cs

BlogRSS Methods

The methods of the BlogRSS class are listed here.

Public Methods

	Name	Description
	Fill (see page 62)	Overrideable method that can be used to grab data from the back end for each server control.

BlogRSS.Fill Method

Overrideable method that can be used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

BlogRSS Properties

The properties of the BlogRSS class are listed here.

Public Properties

	Name	Description
	BlogID (see page 62)	The ID of the blog to be output as an RSS feed.
	CacheInterval (see page 63)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	Text (see page 63)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.
	WrapTag (see page 63)	Allows a developer to specify control's tag. Default is assumed " ".

BlogRSS.BlogID Property

The ID of the blog to be output as an RSS feed.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("\\")]
[Description("The ID of the blog to be output as an RSS feed.")]
[Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BlogID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("\\")>
```

```
<Description("The ID of the blog to be output as an RSS feed.")>
<Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BlogID() As long
```

BlogRSS.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
Public virtual Property CacheInterval() As double
```

BlogRSS.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

```
public override string Text;
```

Visual Basic

```
Public override Property Text() As string
```

BlogRSS.WrapTag Property

Allows a developer to specify control's tag. Default is assumed " ".

C#

```
[Browsable(false)]
[Description("Allows a developer to specify control's tag.")]
[DefaultValue("div")]
public new string WrapTag;
```

Visual Basic

```
<Browsable(false)>
<Description("Allows a developer to specify control's tag.")>
<DefaultValue("div")>
Public new Property WrapTag() As string
```

BreadCrumb Class

The BreadCrumb server control displays history breadcrumbs. Your site visitors can use the breadcrumb trail to jump to previously visited pages.

Class Hierarchy

C#

```
[ DefaultProperty("DisplayTitle"), ToolboxData("<{0}:BreadCrumb
runat=server></{0}:BreadCrumb>") ]
public class BreadCrumb : EkWebControl;
```

Visual Basic

```
< DefaultProperty("DisplayTitle"), ToolboxData("<{0}:BreadCrumb
runat=server></{0}:BreadCrumb>") >
Public Class BreadCrumb
Inherits EkWebControl
```

File

BreadCrumb.cs

BreadCrumb Methods

The methods of the BreadCrumb class are listed here.

Public Methods

	Name	Description
	Fill (see page 64)	Method that can be overridden and used to grab data from the back end for each server control.
	GenerateTrail_HTML (see page 64)	Generates Trail HTML for both vertical and horizontal styles This demonstrates usage of a HtmlTextWriter.
	ResetTrail (see page 64)	This is the overview for the ResetTrail method overload.

BreadCrumb.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

BreadCrumb.GenerateTrail_HTML Method

Generates Trail HTML for both vertical and horizontal styles

This demonstrates usage of a HtmlTextWriter.

C#

```
public string GenerateTrail_HTML();
```

Visual Basic

```
Public Function GenerateTrail_HTML() As string
```

Parameters

Parameters	Description
output	HtmlTextWriter in which to write.

ResetTrail Method

This is the overview for the ResetTrail method overload.

Description

This method is overloaded. The ResetTrail method:

- Resets and removes all current Trail items
- Resets the Trail with an array list of new Trail items
- Resets the Trail to the given Trail item

Overload List

	Name	Description
	BreadCrumb.ResetTrail () (see page 65)	Resets and removes all current Trail items.
	BreadCrumb.ResetTrail (ArrayList) (see page 65)	Resets the trail with an array list of new trail items.
	BreadCrumb.ResetTrail (BreadCrumbItem) (see page 65)	Resets the trail to the given trail item.

BreadCrumb.ResetTrail Method ()

Resets and removes all current Trail items.

C#

```
public void ResetTrail();
```

Visual Basic

```
Public Function ResetTrail() As void
```

BreadCrumb.ResetTrail Method (ArrayList)

Resets the trail with an array list of new trail items.

C#

```
public void ResetTrail(ArrayList TrailData);
```

Visual Basic

```
Public Function ResetTrail(TrailData As ArrayList) As void
```

Parameters

Parameters	Description
ArrayList TrailData	An array list of trail items

BreadCrumb.ResetTrail Method (BreadCrumbItem)

Resets the trail to the given trail item.

C#

```
public void ResetTrail(BreadCrumbItem item);
```

Visual Basic

```
Public Function ResetTrail(item As BreadCrumbItem) As void
```

Parameters

Parameters	Description
BreadCrumbItem item	A trail item.

BreadCrumb Properties

The properties of the BreadCrumb class are listed here.

Public Properties

	Name	Description
	CurrentPageIndicator (see page 66)	If desired, enter one or more symbols or characters that identify the current page in the breadcrumb trail.
	DefaultContentID (see page 67)	Gets or sets the default content ID for this control.
	DisplayStyle (see page 67)	Indicate how to display the breadcrumb trail: horizontally or vertically. The default is Horizontal.
	DisplayTitle (see page 67)	Enter text to describe this Web form when it appears in the breadcrumb trail. For example, if the Web form whose properties you are defining is used for all Human Resources pages on your site, enter Human Resources.
	DynamicParameter (see page 67)	Gets or Sets the QueryString parameter to read a content ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)
	IconAlt (see page 68)	If you define an image in the IconPath (see page 68) property, enter any “Alt” text that should appear when a site visitor hovers the cursor over that image.
	IconPath (see page 68)	If you want the breadcrumb trail to display an image to identify this Web form, enter the path to the image. **Important:** The image location must be relative to the Web root.
	LinkLastItem (see page 68)	Determine whether the last breadcrumb item appears as a hyperlink on this Web form.
	LinkTarget (see page 68)	Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self. • **_Self** = opens in same window • **_Top** = opens in parent window • **_Blank** = opens in new window • **_Parent** = opens in the parent frame
	MaxItems (see page 69)	The maximum number of items in the breadcrumb trail on this Web form. (0 = Unlimited)
	MetadataName (see page 69)	Gets or sets the metadata of the content block for BreadCrumb (see page 63) display title. If not specified it will use content block title.
	Mode (see page 69)	Allows you to make the breadcrumb trail appear as non-hyperlinked plain text. The default is Normal. • **Normal** = breadcrumb trail is hyperlinked • **DisplayOnly** = breadcrumb trail is plain text
	separator (see page 70)	Enter one or more characters to separate the items in a breadcrumb trail on this Web form. The default character is the greater than sign (>).

BreadCrumb.CurrentPageIndicator Property

If desired, enter one or more symbols or characters that identify the current page in the breadcrumb trail.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(""), Description("Set this value to show it
against the current page.") ]
public string CurrentPageIndicator;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(""), Description("Set this value to show it
against the current page.") >
Public Property CurrentPageIndicator() As string
```

BreadCrumb.DefaultContentID Property

Gets or sets the default content ID for this control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the content ID for BreadCrumb display title.")]
public long DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the content ID for BreadCrumb display title.")>
Public Property DefaultContentID() As long
```

BreadCrumb.DisplayStyle Property

Indicate how to display the breadcrumb trail: horizontally or vertically. The default is Horizontal.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(DisplayStyles.Horizontal), Description("The
style of Trail to use."), ]
public new DisplayStyles DisplayStyle;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(DisplayStyles.Horizontal), Description("The
style of Trail to use."), >
Public new Property DisplayStyle() As DisplayStyles
```

BreadCrumb.DisplayTitle Property

Enter text to describe this Web form when it appears in the breadcrumb trail. For example, if the Web form whose properties you are defining is used for all Human Resources pages on your site, enter Human Resources.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(""), Description("Specify the display text
for the current page link in the trail.") ]
public new string DisplayTitle;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(""), Description("Specify the display text
for the current page link in the trail.") >
Public new Property DisplayTitle() As string
```

BreadCrumb.DynamicParameter Property

Gets or Sets the QueryString parameter to read a content ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Content ID dynamically. Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Content ID dynamically. Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

BreadCrumb.IconAlt Property

If you define an image in the IconPath (see page 68) property, enter any “Alt” text that should appear when a site visitor hovers the cursor over that image.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(""), Description("This is the \"alt\" value
that will be assigned to your item icon.") ]
public string IconAlt;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(""), Description("This is the \"alt\" value
that will be assigned to your item icon.") >
Public Property IconAlt() As string
```

BreadCrumb.IconPath Property

If you want the breadcrumb trail to display an image to identify this Web form, enter the path to the image.

Important: The image location must be relative to the Web root.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(""), Description("Use image instead of the
DisplayTitle. This path must be relative to the web root.") ]
public string IconPath;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(""), Description("Use image instead of the
DisplayTitle. This path must be relative to the web root.") >
Public Property IconPath() As string
```

BreadCrumb.LinkLastItem Property

Determine whether the last breadcrumb item appears as a hyperlink on this Web form.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(false), Description("Indicates if the last
item in the Breadcrumb Trail is linked.") ]
public bool LinkLastItem;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(false), Description("Indicates if the last
item in the Breadcrumb Trail is linked.") >
Public Property LinkLastItem() As bool
```

BreadCrumb.LinkTarget Property

Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self.

- **_Self** = opens in same window
- **_Top** = opens in parent window

- **_Blank** = opens in new window
- **_Parent** = opens in the parent frame

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

BreadCrumb.MaxItems Property

The maximum number of items in the breadcrumb trail on this Web form. (0 = Unlimited)

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(5), Description("The maximum number of items to display in the Breadcrumb Trail. (0 = Unlimited)") ]
public int MaxItems;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(5), Description("The maximum number of items to display in the Breadcrumb Trail. (0 = Unlimited)") >
Public Property MaxItems() As Integer
```

BreadCrumb.MetadataName Property

Gets or sets the metadata of the content block for BreadCrumb (see page 63) display title. If not specified it will use content block title.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(" -None -")]
[Editor(typeof(MetaDataUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Description("Gets or Sets the metadata of the content block for BreadCrumb display title. If not specified it will use content block title.")]
public string MetadataName;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(" -None -")>
<Editor(typeof(MetaDataUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("Gets or Sets the metadata of the content block for BreadCrumb display title. If not specified it will use content block title.")>
Public Property MetadataName() As string
```

BreadCrumb.Mode Property

Allows you to make the breadcrumb trail appear as non-hyperlinked plain text. The default is Normal.

- **Normal** = breadcrumb trail is hyperlinked
- **DisplayOnly** = breadcrumb trail is plain text

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(Modes.Normal), Description("Set to \"display only\" mode to render items in the Trail without adding links to it.") ]
public Modes Mode;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(Modes.Normal), Description("Set to
\"display only\" mode to render items in the Trail without adding links to it.") >
Public Property Mode() As Modes
```

BreadCrumb.separator Property

Enter one or more characters to separate the items in a breadcrumb trail on this Web form. The default character is the greater than sign (>).

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(" > "), Description("The separator between
links in the trail.") ]
public string separator;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(" > "), Description("The separator between
links in the trail.") >
Public Property separator() As string
```

BreadCrumbItem Class

Class to contain compass item data.

Class Hierarchy

Ektron.Cms.Controls.BreadCrumbItem

C#

```
public class BreadCrumbItem;
```

Visual Basic

```
Public Class BreadCrumbItem
```

File

BreadCrumbItem.cs

BreadCrumbItem Constructor

This is the overview for the BreadCrumbItem constructor overload.

Overload List

	Name	Description
	BreadCrumbItem.BreadCrumbItem () (see page 70)	Default no parameter constructor.
	BreadCrumbItem.BreadCrumbItem (string, string, string, string, string) (see page 71)	All parameter constructor.

BreadCrumbItem.BreadCrumbItem Constructor ()

Default no parameter constructor.

C#

```
public BreadCrumbItem();
```

Visual Basic

```
Public Sub New()
```

BreadCrumbItem.BreadCrumbItem Constructor (string, string, string, string, string)

All parameter constructor.

C#

```
public BreadCrumbItem(string rawUrl, string displayTitle, string iconPath, string iconAlt,
string target);
```

Visual Basic

```
Public Sub New(rawUrl As string, displayTitle As string, iconPath As string, iconAlt As
string, target As string)
```

Parameters

Parameters	Description
string rawUrl	The URL to use for the item link.
string displayTitle	The title for use with the display in the compass.
string iconPath	A root relative path to the item icon.
string iconAlt	Text to use as the item alt text.
string target	The target of the item.

BreadCrumbItem Methods

The methods of the BreadCrumbItem class are listed here.

Public Methods

	Name	Description
	Equals (see page 71)	Overridden equals function returns false on null object comparison. And, when the given object and current object .ToString (see page 72)() do not match, The ToString (see page 72)() function is also overridden to allow this to work.
	GetHashCode (see page 71)	Overridden otherwise VS.NET gives a warning.
	ToString (see page 72)	Returns a concatenated string of all class variables.ToString().

BreadCrumbItem.Equals Method

Overridden equals function returns false on null object comparison. And, when the given object and current object .ToString (see page 72)() do not match, The ToString (see page 72)() function is also overridden to allow this to work.

C#

```
public override bool Equals(object obj);
```

Visual Basic

```
Public override Function Equals(obj As object) As bool
```

Parameters

Parameters	Description
object obj	_nt_

Returns

BreadCrumbItem.GetHashCode Method

Overridden otherwise VS.NET gives a warning.

C#

```
public override int GetHashCode();
```

Visual Basic

```
Public override Function GetHashCode() As Integer
```

Returns

BreadCrumbItem.ToString Method

Returns a concatenated string of all class variables.ToString().

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

BreadCrumbItem Properties

The properties of the BreadCrumbItem class are listed here.

Public Properties

	Name	Description
S	CaseSensitiveUrlComparison (see page 72)	Gets / Sets the comparison mode to use when using .equals. This is also used by ArrayList.Contains(). This is static to allow it to be set before any comparison takes place.
	DisplayTitle (see page 72)	Gets / Sets the title for use with the display in the compass.
	IconAlt (see page 72)	Gets / Sets the text to use as the item alt text.
	IconPath (see page 73)	Gets / Sets the root relative path to the item icon.
	RawUrl (see page 73)	Gets / Sets the URL to use for the item link.
	Target (see page 73)	Gets / Sets the target of the item.

BreadCrumbItem.CaseSensitiveUrlComparison Property

Gets / Sets the comparison mode to use when using .equals. This is also used by ArrayList.Contains().

This is static to allow it to be set before any comparison takes place.

C#

```
public static bool CaseSensitiveUrlComparison;
```

Visual Basic

```
Public static Property CaseSensitiveUrlComparison() As bool
```

BreadCrumbItem.DisplayTitle Property

Gets / Sets the title for use with the display in the compass.

C#

```
public string DisplayTitle;
```

Visual Basic

```
Public Property DisplayTitle() As string
```

BreadCrumbItem.IconAlt Property

Gets / Sets the text to use as the item alt text.

C#

```
public string IconAlt;
```

Visual Basic

```
Public Property IconAlt() As string
```

BreadCrumbItem.IconPath Property

Gets / Sets the root relative path to the item icon.

C#

```
public string IconPath;
```

Visual Basic

```
Public Property IconPath() As string
```

BreadCrumbItem.RawUrl Property

Gets / Sets the URL to use for the item link.

C#

```
public string RawUrl;
```

Visual Basic

```
Public Property RawUrl() As string
```

BreadCrumbItem.Target Property

Gets / Sets the target of the item.

C#

```
public string Target;
```

Visual Basic

```
Public Property Target() As string
```

BusinessRules Class

Class Hierarchy

C#

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:BusinessRules runat=server></{0}:BusinessRules>")]
public class BusinessRules : EkWebControl;
```

Visual Basic

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:BusinessRules runat=server></{0}:BusinessRules>")>
Public Class BusinessRules
Inherits EkWebControl
```

File

BusinessRules.cs

Description

Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example:

- **Condition** - a site visitor visits your site for the first time
- **Action** - add a cookie to the site visitor's system and pop up a welcome message

With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control to your Web form, and assign the server control the Business Rules Ruleset.

.NET Assemblies BusinessRule API vs. BusinessRules Server Control API

The .NET Assemblies BusinessRule (see page 671) API (see page 669) allows you to work with results of a Business Rule. The BusinessRules server control API (see page 669) allows you to customize how and where a Business Rule is used on a Web page.

To learn about the .NET Assemblies BusinessRule (see page 671) API (see page 669), click here (see page 671).

BusinessRules Methods

The methods of the BusinessRules class are listed here.

Public Methods

	Name	Description
	Fill (see page 74)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 74)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

BusinessRules.Fill Method

Overrideable method that can be used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

BusinessRules.Wrap Method

Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

BusinessRules Properties

The properties of the BusinessRules class are listed here.

Public Properties

	Name	Description
	Parameters (see page 75)	User (see page 950) Defined parameters that can be evaluated in rules.
	Result (see page 75)	An array of boolean values representing the results of the evaluations.
	RuleID (see page 75)	The ID number of the Ruleset to evaluate.
	Text (see page 76)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.
	WrapTag (see page 76)	Overridden and hidden - should not be wrapped by span

BusinessRules.Parameters Property

User (see page 950) Defined parameters that can be evaluated in rules.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Localizable(true)]
[Description("User Defined parameters that can be evaluated in rules.")]
[Browsable(false)]
public System.Collections.Specialized.NameValueCollection Parameters;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Localizable(true)>
<Description("User Defined parameters that can be evaluated in rules.")>
<Browsable(false)>
Public Property Parameters() As System.Collections.Specialized.NameValueCollection
```

BusinessRules.Result Property

An array of boolean values representing the results of the evaluations.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Localizable(true)]
[Description("An array of boolean values representing the results of the evaluations.")]
public bool Result;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Localizable(true)>
<Description("An array of boolean values representing the results of the evaluations.")>
Public Property Result() As bool
```

BusinessRules.RuleID Property

The ID number of the Ruleset to evaluate.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Localizable(true)]
[Description("The ID number of the Ruleset to evaluate.")]
[Editor(typeof(RulesetUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long RuleID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Localizable(true)>
<Description("The ID number of the Ruleset to evaluate.")>
<Editor(typeof(RulesetUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property RuleID() As long
```

BusinessRules.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

```
public override string Text;
```

Visual Basic

```
Public override Property Text() As string
```

BusinessRules.WrapTag Property

Overridden and hidden - should not be wrapped by span

C#

```
[Browsable(false)]
public new string WrapTag;
```

Visual Basic

```
<Browsable(false)>
Public new Property WrapTag() As string
```

Calendar Class

Summary description for Calendar.

Class Hierarchy

C#

```
[DefaultProperty("DefaultCalendarID"), ToolboxData("<{0}:Calendar
runat=server></{0}:Calendar>")]
public class Calendar : EkWebPart;
```

Visual Basic

```
<DefaultProperty("DefaultCalendarID"), ToolboxData("<{0}:Calendar
runat=server></{0}:Calendar>")>
Public Class Calendar
Inherits EkWebPart
```

File

Calendar.cs

Calendar Methods

The methods of the Calendar class are listed here.

Public Methods

	Name	Description
	Fill (see page 77)	Method that can be overridden and used to grab data from the back end for each server control.
	Wrap (see page 77)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Calendar.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Calendar.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

Calendar Properties

The properties of the Calendar class are listed here.

Public Properties

	Name	Description
	DefaultCalendarID (see page 78)	The default ID of the Calendar (see page 76) to be displayed.
	DisplayType (see page 78)	Specify the the Calendar (see page 76) view. Choices are: • "Month" • "Day" • "Inline" • "MonthUpWithEvent"
	DynamicParameter (see page 78)	The QueryString to check for a dynamic ID.
	EndDate (see page 79)	Specify the end date the Calendar (see page 76) displays.
	HighlightEventID (see page 79)	Specify the event ID which the Calendar (see page 76) will highlight.
	HighlightEventTypeID (see page 79)	Specify the event type and the Calendar (see page 76) highlights that type of event(s).
	ShowXML (see page 80)	Text (see page 80) property will hold raw calendar XML. The Calendar (see page 76) is not visible at runtime.
	StartDate (see page 80)	Specify the start date the calendar displays.

	Text (see page 80)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

Calendar.DefaultCalendarID Property

The default ID of the Calendar (see page 76) to be displayed.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Editor(typeof(CalendarUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Description("The default ID of the Calendar to be displayed.")]
[Personalizable()]
public long DefaultCalendarID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Editor(typeof(CalendarUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("The default ID of the Calendar to be displayed.")>
<Personalizable()>
Public Property DefaultCalendarID() As long
```

Calendar.DisplayType Property

Specify the the Calendar (see page 76) view. Choices are:

- "Month"
- "Day"
- "Inline"
- "MonthUpWithEvent"

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Month")]
[TypeConverter(typeof(CalXsltTypeConverter))]
[Description("Specify the the Calendar view.")]
[Personalizable()]
public string DisplayType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Month")>
<TypeConverter(typeof(CalXsltTypeConverter))>
<Description("Specify the the Calendar view.")>
<Personalizable()>
Public Property DisplayType() As string
```

Calendar.DynamicParameter Property

The QueryString to check for a dynamic ID.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The QueryString to check for a dynamic ID.")]
[TypeConverter(typeof(DynamicCalendarTypeConverter))]
[DefaultValue("None - Use Default")]
```

```
[Personalizable()]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The QueryString to check for a dynamic ID.")>
<TypeConverter(typeof(DynamicCalendarTypeConverter))>
<DefaultValue("None - Use Default")>
<Personalizable()>
Public Property DynamicParameter() As string
```

Calendar.EndDate Property

Specify the end date the Calendar (see page 76) displays.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify the end date the Calendar displays.")]
[Personalizable(), WebBrowsable(true), WebDisplayName("End Date"), WebDescription("Specify the
end date the Calendar displays.")]
public DateTime EndDate;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify the end date the Calendar displays.")>
<Personalizable(), WebBrowsable(true), WebDisplayName("End Date"), WebDescription("Specify the
end date the Calendar displays.")>
Public Property EndDate() As DateTime
```

Calendar.HighlightEventID Property

Specify the event ID which the Calendar (see page 76) will highlight.

C#

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Specify the event ID which the Calendar will highlight.")]
[Personalizable()]
public long HighlightEventID;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Specify the event ID which the Calendar will highlight.")>
<Personalizable()>
Public Property HighlightEventID() As long
```

Calendar.HighlightEventTypeID Property

Specify the event type and the Calendar (see page 76) highlights that type of event(s).

C#

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Specify the event type and the Calendar highlights that type of event(s).")]
[Personalizable()]
public long HighlightEventTypeID;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Specify the event type and the Calendar highlights that type of event(s).")>
<Personalizable()>
Public Property HighlightEventTypeID() As long
```

Calendar.ShowXML Property

Text (see page 80) property will hold raw calendar XML. The Calendar (see page 76) is not visible at runtime.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Text property will hold raw calendar XML. The Calendar is not visible at
runtime.")]
[Personalizable()]
public bool ShowXML;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Text property will hold raw calendar XML. The Calendar is not visible at
runtime.")>
<Personalizable()>
Public Property ShowXML() As bool
```

Calendar.StartDate Property

Specify the start date the calendar displays.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify the start date the Calendar displays.")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Start Date"), WebDescription("Specify
the start date the Calendar displays.")]
public DateTime StartDate;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify the start date the Calendar displays.")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Start Date"), WebDescription("Specify
the start date the Calendar displays.")>
Public Property StartDate() As DateTime
```

Calendar.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

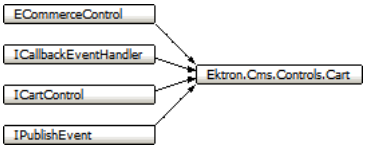
```
public override string Text;
```

Visual Basic

```
Public override Property Text() As string
```

Cart Class

Class Hierarchy



C#

```
[ToolboxData("<{0}:Cart runat=server></{0}:Cart>")]
public class Cart : ECommerceControl, ICallbackEventHandler, ICartControl, IPublishEvent;
```

Visual Basic

```
<ToolboxData("<{0}:Cart runat=server></{0}:Cart>")>
Public Class Cart
Inherits ECommerceControl
Inherits ICallbackEventHandler
Inherits ICartControl
Inherits IPublishEvent
```

File

Cart.cs

Description

This is class Ektron.Cms.Controls.Cart.

Cart Properties

The properties of the Cart class are listed here.

Public Properties

	Name	Description
	DisplayXslt (see page 82)	Specify an external XSLT file to render the control. By default, the control uses Cart.xsl. This file is located in:
	DynamicProductParameter (see page 82)	The QueryString parameter name which is used to pass the product ID to the product details page. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors clicks a product's title, this parameter is passed with the product's ID to the product details page.
	EnableCoupons (see page 82)	Set to true to allow a site visitor to enter coupon codes for discounts. • **True** = coupon entry is available. • **False** = coupon entry is not available.
	LoadingImage (see page 83)	Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**.
	Settings (see page 83)	The application settings.
	Stylesheet (see page 83)	Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.
	TemplateCheckout (see page 84)	The template location that contains the Checkout (see page 85) server control. This is the page a site visitor is sent to when the he clicks the Checkout (see page 85) button. If the template file is located in the same folder as the Web form containing this server control, just enter its name. Otherwise, the path to the file can be relative or absolute.
	TemplateProduct (see page 84)	The template location that contains the Product (see page 431) server control. This is the page a site visitor is sent to when he clicks the name of a product in his cart. This page provides details about the product. When this property is left blank, the server control uses the Product (see page 431)'s QuickLink information. If the template file is located in the same folder as the Web form containing this server control, just enter its name. Otherwise, the path to the file can be relative or absolute.
	TemplateShopping (see page 85)	The URL to navigate to when the site visitor clicks the Continue Shopping button. This URL could lead to a template containing a ProductSearch (see page 445) or ProductList (see page 437) server control that would allow a site visitor to select additional products.

	UserControlUrl (see page 85)	The URL of the User (see page 950) Control that manages the display of the Cart (see page 81). This path can be relative or absolute. A sample user control is located in:

Cart.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses Cart.xsl. This file is located in:

```
[Site Root]\Workarea\Xslt\Commerce
```

Important: If you want to edit this file, it is strongly recommended that you make a copy, change its name and move it outside of the Workarea folder. If you make changes to this file and do not move it out of the Workarea, this file and the changes will be lost when you upgrade.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses \"Cart.xsl.\"")]
[TypeConverter(typeof(XsltCartTypeConverter))]
[DefaultValue("Cart.xsl")]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses \"Cart.xsl.\"")>
<TypeConverter(typeof(XsltCartTypeConverter))>
<DefaultValue("Cart.xsl")>
Public override Property DisplayXslt() As string
```

Cart.DynamicProductParameter Property

The QueryString parameter name which is used to pass the product ID to the product details page. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors clicks a product's title, this parameter is passed with the product's ID to the product details page.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The QueryString parameter name which is used to pass the product ID to the
product details page. For example, if your QueryString parameter for products is PID, enter
that in this property. Then, when site visitors clicks a product?s title, this parameter is
passed with the product?s ID to the product details page.")]
[DefaultValue("id")]
public string DynamicProductParameter;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The QueryString parameter name which is used to pass the product ID to the
product details page. For example, if your QueryString parameter for products is PID, enter
that in this property. Then, when site visitors clicks a product?s title, this parameter is
passed with the product?s ID to the product details page.")>
<DefaultValue("id")>
Public Property DynamicProductParameter() As string
```

Cart.EnableCoupons Property

Set to true to allow a site visitor to enter coupon codes for discounts.

- **True** = coupon entry is available.

- **False** = coupon entry is not available.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Set to True to allow a site visitor to enter coupon codes for discounts.")]
[DefaultValue("true")]
public bool EnableCoupons;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Set to True to allow a site visitor to enter coupon codes for discounts.")>
<DefaultValue("true")>
Public Property EnableCoupons() As bool
```

Cart.LoadingImage Property

Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")]
[DefaultValue("ajax-loader_circle_lg.gif")]
public string LoadingImage;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")>
<DefaultValue("ajax-loader_circle_lg.gif")>
Public Property LoadingImage() As string
```

Cart.Settings Property

The application settings.

C#

```
[Category("Ektron")]
[Description("The application settings.")]
[Browsable(false)]
public SettingsData Settings;
```

Visual Basic

```
<Category("Ektron")>
<Description("The application settings.")>
<Browsable(false)>
Public Property Settings() As SettingsData
```

Cart.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[TypeConverter(typeof(CartCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/Cart.css")]
```

```
public string Stylesheet;
```

Visual Basic

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<TypeConverter(typeof(CartCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/Cart.css")>
Public Property Stylesheet() As string
```

Cart.TemplateCheckout Property

The template location that contains the Checkout (see page 85) server control. This is the page a site visitor is sent to when the he clicks the Checkout (see page 85) button. If the template file is located in the same folder as the Web form containing this server control, just enter its name. Otherwise, the path to the file can be relative or absolute.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The template location that contains the Checkout server control. This is the
page a site visitor is sent to when the he clicks the Checkout button. If the template file is
located in the same folder as the Web form containing this server control, just enter its
name. Otherwise, the path to the file can be relative or absolute.")]
[DefaultValue("")]
public string TemplateCheckout;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The template location that contains the Checkout server control. This is the
page a site visitor is sent to when the he clicks the Checkout button. If the template file is
located in the same folder as the Web form containing this server control, just enter its
name. Otherwise, the path to the file can be relative or absolute.")>
<DefaultValue("")>
Public Property TemplateCheckout() As string
```

Cart.TemplateProduct Property

The template location that contains the Product (see page 431) server control. This is the page a site visitor is sent to when he clicks the name of a product in his cart. This page provides details about the product. When this property is left blank, the server control uses the Product (see page 431)'s QuickLink information. If the template file is located in the same folder as the Web form containing this server control, just enter its name. Otherwise, the path to the file can be relative or absolute.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The template location that contains the Product server control. This is the page
a site visitor is sent to when he clicks the name of a product in his cart. This page provides
details about the product.")]
[DefaultValue("")]
public string TemplateProduct;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The template location that contains the Product server control. This is the page
a site visitor is sent to when he clicks the name of a product in his cart. This page provides
details about the product.")>
<DefaultValue("")>
Public Property TemplateProduct() As string
```

Cart.TemplateShopping Property

The URL to navigate to when the site visitor clicks the Continue Shopping button. This URL could lead to a template containing a ProductSearch (see page 445) or ProductList (see page 437) server control that would allow a site visitor to select additional products.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The URL to navigate to, when the user clicks the 'Continue Shopping' button.")]
[DefaultValue("")]
public string TemplateShopping;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The URL to navigate to, when the user clicks the 'Continue Shopping' button.")>
<DefaultValue("")>
Public Property TemplateShopping() As string
```

Cart.UserControlUrl Property

The URL of the User (see page 950) Control that manages the display of the Cart (see page 81). This path can be relative or absolute. A sample user control is located in:

```
[siteroot]\Workarea\ControlTemplates\Commerce\Cart\CartUserControl.ascx.
```

C#

```
[Category("Ektron")]
[Description("The URL of the User Control that manages the display of the Cart. This path can
be relative or absolute. A sample user control is located in
<siteroot>\\Workarea\\ControlTemplates\\Commerce\\Cart\\CartUserControl.ascx.")]
[TypeConverter(typeof(string))]
[DefaultValue("")]
public string UserControlUrl;
```

Visual Basic

```
<Category("Ektron")>
<Description("The URL of the User Control that manages the display of the Cart. This path can
be relative or absolute. A sample user control is located in
<siteroot>\\Workarea\\ControlTemplates\\Commerce\\Cart\\CartUserControl.ascx.")>
<TypeConverter(typeof(string))>
<DefaultValue("")>
Public Property UserControlUrl() As string
```

Checkout Class

Class Hierarchy

C#

```
[DefaultProperty("DefaultCheckoutID"), ToolboxData("<{0}:Checkout
runat=server></{0}:Checkout>")]
public class Checkout : ECommerceControl, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultCheckoutID"), ToolboxData("<{0}:Checkout
runat=server></{0}:Checkout>")>
Public Class Checkout
```

```
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

File

Checkout.cs

Description

This is class Ektron.Cms.Controls.Checkout.

Checkout Enumerations

The enumerations of the Checkout class are listed here.

Enumerations

	Name	Description
	ControlPhase (see page 86)	This is record Ektron.Cms.Controls.Checkout.ControlPhase.

Ektron.Cms.Controls.Checkout.ControlPhase Enumeration

C#

```
public enum ControlPhase {
}
```

Visual Basic

```
Public Enum ControlPhase
End Enum
```

File

Checkout.cs

Description

This is record Ektron.Cms.Controls.Checkout.ControlPhase.

Checkout Methods

The methods of the Checkout class are listed here.

Public Methods

	Name	Description
	Fill (see page 86)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 86)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 87)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Checkout.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Checkout.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Checkout.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

Checkout Properties

The properties of the Checkout class are listed here.

Public Properties

	Name	Description
	Cart (see page 88)	Provides access to the current Cart (Basket (see page 1437)) object.
	CurrentPhase (see page 88)	Reports or selects the current page (or phase) of the control, particularly useful if wizard mode is enabled.
	Customer (see page 88)	Provides access to the current CustomerData (see page 1478) object.
	DefaultCountryId (see page 89)	The ID of the default country that appears in the Billing Address and Shipping Address. Set this ID to the country from which the majority of your customers will be making purchases. If the site visitor is from a different country, he can change it when editing the Billing or Shipping Address.
	DisplayXslt (see page 89)	Specify an external XSLT file to render the control. By default, the control uses Checkout.xsl. This file is located in:
	DynamicPhaseParameter (see page 89)	The name of the parameter on the query string that identifies desired phase ID.
	FriendlyErrorMessage (see page 90)	The message that is shown when an unhandled error occurs. Details are sent to the event log. The default message is
	IsSSLRequired (see page 90)	When set to True, switches to an SSL encrypted URL. For the SSL encryption to work, you must have an SSL certificate installed. Information on SSL can be found at the following Web sites: • http://www.tech-faq.com/understanding-ssl.shtml • http://www.verisign.com/ssl/ssl-information-center/how-ssl-security-works/index.html **Note:** Ektron, Inc. is not associated with TechFAQ or VeriSign. However, both these sites offer very good, detailed explanations of Secure Sockets Layer.
	LoadingImage (see page 91)	The image to display while the control is fetching data. The default is:
	PaymentMethod (see page 91)	Specifies the class to call to collect the payment information. Change this to accept other types of payments, such as eChecks.
	ShowErrorDetails (see page 91)	When true, displays detailed error message, otherwise shows friendly error message (default).
	Stylesheet (see page 92)	Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.
	TemplateCart (see page 92)	The URL path of the template that contains the Cart (see page 88) server control. This path can be relative or absolute. When a path is entered, a link appears in the Review Order part of the process that allows a user to navigate back to the template containing the Cart (see page 88) server control.

	TemplateOrderHistory (see page 92)	The URL path of the template that contains the OrderList (see page 414) server control. This path can be relative or absolute. When a path is entered, a link appears at the end of checkout process that allows a user to navigate to the template containing the OrderList (see page 414) server control.
	TemplateRecoverPassword (see page 93)	The URL path of the template that helps site visitors recover their password. This path can be relative or absolute. When a path is entered and a site visitor has not logged in, a link appears at the beginning of checkout process that allows him to navigate to a template containing information to recover his password. The Membership (see page 360) server control contains an option for recovering passwords.
	TemplateShopping (see page 93)	The URL path of the template that allows the site visitor to continue shopping. When a path is entered, a link appears at the end of checkout process that allows a user to navigate to the template that would allow them to continue shopping; for example, the template containing a ProductList (see page 437) or ProductSearch (see page 445) server control.
	UseMsAjax (see page 94)	If true, uses Microsoft Ajax to perform callback. Otherwise, the control uses Ektron Library (see page 695) callback (default).
	ValidatePhase (see page 94)	When true (default) ensures current phase is allowed. Set to false to override, allowing any phase order.

Checkout.Cart Property

Provides access to the current Cart (Basket (see page 1437)) object.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(Ektron.Cms.Commerce.Basket))]
[Description("")]
[Bindable(true)]
[DefaultValue("null")]
public Ektron.Cms.Commerce.Basket Cart;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(Ektron.Cms.Commerce.Basket))>
<Description("")>
<Bindable(true)>
<DefaultValue("null")>
Public Property Cart() As Ektron.Cms.Commerce.Basket
```

Checkout.CurrentPhase Property

Reports or selects the current page (or phase) of the control, particularly useful if wizard mode is enabled.

C#

```
[Category("Ektron")]
[Browsable(false)]
[TypeConverter(typeof(ControlPhase))]
[Description("Reports or selects the current page (or phase) of the control, particularly
useful if wizard mode is enabled.")]
[DefaultValue("")]
public ControlPhase CurrentPhase;
```

Visual Basic

```
<Category("Ektron")>
<Browsable(false)>
<TypeConverter(typeof(ControlPhase))>
<Description("Reports or selects the current page (or phase) of the control, particularly
useful if wizard mode is enabled.")>
<DefaultValue("")>
Public Property CurrentPhase() As ControlPhase
```

Checkout.Customer Property

Provides access to the current CustomerData (see page 1478) object.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(Ektron.Cms.Commerce.Basket))]
```

```
[Description("")]
[Bindable(true)]
[DefaultValue("null")]
public Ektron.Cms.Commerce.CustomerData Customer;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(Ektron.Cms.Commerce.Basket))>
<Description("")>
<Bindable(true)>
<DefaultValue("null")>
Public Property Customer() As Ektron.Cms.Commerce.CustomerData
```

Checkout.DefaultCountryId Property

The ID of the default country that appears in the Billing Address and Shipping Address. Set this ID to the country from which the majority of your customers will be making purchases. If the site visitor is from a different country, he can change it when editing the Billing or Shipping Address.

C#

```
[Category("Ektron")]
[Description("The ID of the default country that appears in the Billing Address and Shipping
Address. Set this ID to the country from which the majority of your customers will be making
purchases. If the site visitor is from a different country, he can change it when editing the
Billing or Shipping Address..")]
public int DefaultCountryId;
```

Visual Basic

```
<Category("Ektron")>
<Description("The ID of the default country that appears in the Billing Address and Shipping
Address. Set this ID to the country from which the majority of your customers will be making
purchases. If the site visitor is from a different country, he can change it when editing the
Billing or Shipping Address..")>
Public Property DefaultCountryId() As Integer
```

Checkout.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses Checkout.xsl. This file is located in:

```
[SiteRoot]\Workarea\Xslt\Commerce\Checkout\Standard
```

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses Checkout.xsl.")]
[TypeConverter(typeof(XsltCheckoutTypeConverter))]
[DefaultValue("Checkout.xsl")]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses Checkout.xsl.")>
<TypeConverter(typeof(XsltCheckoutTypeConverter))>
<DefaultValue("Checkout.xsl")>
Public override Property DisplayXslt() As string
```

Checkout.DynamicPhaseParameter Property

The name of the parameter on the query string that identifies desired phase ID.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("The name of the parameter on the query string that identifies desired phase
ID.")]
[DefaultValue("phase")]
public string DynamicPhaseParameter;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("The name of the parameter on the query string that identifies desired phase
ID.")>
<DefaultValue("phase")>
Public Property DynamicPhaseParameter() As string
```

Checkout.FriendlyErrorMessage Property

The message that is shown when an unhandled error occurs. Details are sent to the event log. The default message is

```
We're sorry, an error occurred while processing your request. Please try again later...
```

C#

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("The message that is shown when an unhandled error occurs. Details are sent to
the event log. The default message is \"We're sorry, an error occurred while processing your
request. Please try again later...\"")]
[DefaultValue("We're sorry, an error occurred...")]
public string FriendlyErrorMessage;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("The message that is shown when an unhandled error occurs. Details are sent to
the event log. The default message is \"We're sorry, an error occurred while processing your
request. Please try again later...\"")>
<DefaultValue("We're sorry, an error occurred...")>
Public Property FriendlyErrorMessage() As string
```

Checkout.IsSSLRequired Property

When set to True, switches to an SSL encrypted URL. For the SSL encryption to work, you must have an SSL certificate installed. Information on SSL can be found at the following Web sites:

- http://www.tech-faq.com/understanding-ssl.shtml
- http://www.verisign.com/ssl/ssl-information-center/how-ssl-security-works/index.html

Note: Ektron, Inc. is not associated with TechFAQ or VeriSign. However, both these sites offer very good, detailed explanations of Secure Sockets Layer.

C#

```
[Category("Ektron")]
[Description("When set to True, switches to an SSL encrypted URL. For the SSL encryption to
work, you must have an SSL certificate installed.")]
[TypeConverter(typeof(bool))]
[DefaultValue(false)]
public bool IsSSLRequired;
```

Visual Basic

```
<Category("Ektron")>
<Description("When set to True, switches to an SSL encrypted URL. For the SSL encryption to
work, you must have an SSL certificate installed.")>
```

```
<TypeConverter(typeof(bool))>
<DefaultValue(false)>
Public Property IsSSLRequired() As bool
```

Checkout.LoadingImage Property

The image to display while the control is fetching data. The default is:

```
[siteroot]\Workarea\images\application\ajax-loader_circle_lg.gif.
```

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The image to display while the control is fetching data. The default is:
[siteroot]\\Workarea\\images\\application\\ajax-loader_circle_lg.gif.")]
[DefaultValue("ajax-loader_circle_lg.gif")]
public string LoadingImage;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The image to display while the control is fetching data. The default is:
[siteroot]\\Workarea\\images\\application\\ajax-loader_circle_lg.gif.")>
<DefaultValue("ajax-loader_circle_lg.gif")>
Public Property LoadingImage() As string
```

Checkout.PaymentMethod Property

Specifies the class to call to collect the payment information. Change this to accept other types of payments, such as eChecks.

C#

```
[Category("Ektron")]
[Browsable(false)]
[Description("Specifies the PaymentMethod used for the order. Change this to accept other
types of payments, such as eChecks.")]
public PaymentMethod PaymentMethod;
```

Visual Basic

```
<Category("Ektron")>
<Browsable(false)>
<Description("Specifies the PaymentMethod used for the order. Change this to accept other
types of payments, such as eChecks.")>
Public Property PaymentMethod() As PaymentMethod
```

Checkout.ShowErrorDetails Property

When true, displays detailed error message, otherwise shows friendly error message (default).

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("When true, displays detailed error message, otherwise shows friendly error
message (default).")]
[DefaultValue("false")]
public bool ShowErrorDetails;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("When true, displays detailed error message, otherwise shows friendly error
message (default).")>
<DefaultValue("false")>
```

```
Public Property ShowErrorDetails() As bool
```

Checkout.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[TypeConverter(typeof(CheckoutCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/Checkout.css")]
public string Stylesheet;
```

Visual Basic

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<TypeConverter(typeof(CheckoutCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/Checkout.css")>
Public Property Stylesheet() As string
```

Checkout.TemplateCart Property

The URL path of the template that contains the Cart (see page 88) server control. This path can be relative or absolute. When a path is entered, a link appears in the Review Order part of the process that allows a user to navigate back to the template containing the Cart (see page 88) server control.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("The URL path of the template that contains the Cart server control. This path
can be relative or absolute. When a path is entered, a link appears in the Review Order part
of the process that allows a user to navigate back to the template containing the Cart server
control.")]
[DefaultValue("")]
public string TemplateCart;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("The URL path of the template that contains the Cart server control. This path
can be relative or absolute. When a path is entered, a link appears in the Review Order part
of the process that allows a user to navigate back to the template containing the Cart server
control.")>
<DefaultValue("")>
Public Property TemplateCart() As string
```

Checkout.TemplateOrderHistory Property

The URL path of the template that contains the OrderList (see page 414) server control. This path can be relative or absolute. When a path is entered, a link appears at the end of checkout process that allows a user to navigate to the template containing the OrderList (see page 414) server control.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("The URL path of the template that contains the OrderList server control. This
path can be relative or absolute. When a path is entered, a link appears at the end of
checkout process that allows a user to navigate to the template containing the OrderList
server control.")]
```

```
[DefaultValue("")]
public string TemplateOrderHistory;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("The URL path of the template that contains the OrderList server control. This
path can be relative or absolute. When a path is entered, a link appears at the end of
checkout process that allows a user to navigate to the template containing the OrderList
server control.")>
<DefaultValue("")>
Public Property TemplateOrderHistory() As string
```

Checkout.TemplateRecoverPassword Property

The URL path of the template that helps site visitors recover their password. This path can be relative or absolute. When a path is entered and a site visitor has not logged in, a link appears at the beginning of checkout process that allows him to navigate to a template containing information to recover his password. The Membership (see page 360) server control contains an option for recovering passwords.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("The URL path of the template that helps site visitors recover their password.
This path can be relative or absolute. When a path is entered and a site visitor has not
logged in, a link appears at the beginning of checkout process that allows him to navigate to
a template containing information to recover his password. The Membership server control
contains an option for recovering passwords.")]
[DefaultValue("")]
public string TemplateRecoverPassword;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("The URL path of the template that helps site visitors recover their password.
This path can be relative or absolute. When a path is entered and a site visitor has not
logged in, a link appears at the beginning of checkout process that allows him to navigate to
a template containing information to recover his password. The Membership server control
contains an option for recovering passwords.")>
<DefaultValue("")>
Public Property TemplateRecoverPassword() As string
```

Checkout.TemplateShopping Property

The URL path of the template that allows the site visitor to continue shopping. When a path is entered, a link appears at the end of checkout process that allows a user to navigate to the template that would allow them to continue shopping; for example, the template containing a ProductList (see page 437) or ProductSearch (see page 445) server control.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("The URL path of the template that allows the site visitor to continue shopping.
When a path is entered, a link appears at the end of checkout process that allows a user to
navigate to the template that would allow them to continue shopping; for example, the template
containing a ProductList or ProductSearch server control.")]
[DefaultValue("")]
public string TemplateShopping;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("The URL path of the template that allows the site visitor to continue shopping.
When a path is entered, a link appears at the end of checkout process that allows a user to
```

```
navigate to the template that would allow them to continue shopping; for example, the template
containing a ProductList or ProductSearch server control.")>
<DefaultValue("")>
Public Property TemplateShopping() As string
```

Checkout.UseMsAjax Property

If true, uses Microsoft Ajax to perform callback. Otherwise, the control uses Ektron Library (see page 695) callback (default).

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("If true, uses Microsoft Ajax to perform callback. Otherwise, the control uses
Ektron Library callback (default).")]
[DefaultValue("false")]
public bool UseMsAjax;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("If true, uses Microsoft Ajax to perform callback. Otherwise, the control uses
Ektron Library callback (default).")>
<DefaultValue("false")>
Public Property UseMsAjax() As bool
```

Checkout.ValidatePhase Property

When true (default) ensures current phase is allowed. Set to false to override, allowing any phase order.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("When true (default) ensures current phase is allowed. Set to false to override,
allowing any phase order")]
[DefaultValue("true")]
public bool ValidatePhase;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("When true (default) ensures current phase is allowed. Set to false to override,
allowing any phase order")>
<DefaultValue("true")>
Public Property ValidatePhase() As bool
```

Checkout Delegates

The delegates of the Checkout class are listed here.

Public Delegates

Name	Description
PhaseChangeEventHandler (see page 94)	For the PhaseChange (see page 95) event.
PostLoginEventHandler (see page 95)	For the Post-Login (see page 347) event.

Checkout.PhaseChangeEventHandler Delegate

For the PhaseChange (see page 95) event.

C#

```
public delegate void PhaseChangeEventHandler(object sender, PhaseChangeEventArgs e);
```

Visual Basic

```
Public Delegate Sub PhaseChangeEventHandler(object sender, PhaseChangeEventArgs e) As void
```

Parameters

Parameters	Description
sender	
e	

Checkout.PostLoginEventHandler Delegate

For the Post-Login (see page 347) event.

C#

```
public delegate void PostLoginEventHandler(object sender, PostLoginEventArgs e);
```

Visual Basic

```
Public Delegate Sub PostLoginEventHandler(object sender, PostLoginEventArgs e) As void
```

Parameters

Parameters	Description
sender	
e	

Checkout Events

The events of the Checkout class are listed here.

Public Events

	Name	Description
	PhaseChange (see page 95)	Fires when the current phase (state) is about to change. Allows supervising phase-changes, in order to allow or prevent changing as needed.
	PostLogin (see page 95)	Fires when an Login (see page 347) event occurs. Allows inspecting or modifying callback parameters.

Checkout.PhaseChange Event

Fires when the current phase (state) is about to change. Allows supervising phase-changes, in order to allow or prevent changing as needed.

C#

```
public event PhaseChangeEventHandler PhaseChange;
```

Visual Basic

```
Public Event PhaseChange As PhaseChangeEventHandler
```

Checkout.PostLogin Event

Fires when an Login (see page 347) event occurs. Allows inspecting or modifying callback parameters.

C#

```
public event PostLoginEventHandler PostLogin;
```

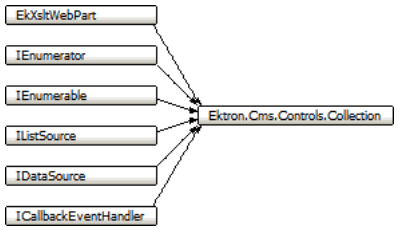
Visual Basic

```
Public Event PostLogin As PostLoginEventHandler
```

Collection Class

Ektron Collection class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.

Class Hierarchy



C#

```
[DefaultProperty("DefaultCollectionID"), ToolboxData("<{0}:Collection
runat=server></{0}:Collection>"), PersistChildren(true)]
public class Collection : EkXsltWebPart, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultCollectionID"), ToolboxData("<{0}:Collection
runat=server></{0}:Collection>"), PersistChildren(true)>
Public Class Collection
Inherits EkXsltWebPart
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
Inherits ICallbackEventHandler
```

File

Collection.cs

Collection Methods

The methods of the Collection class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 96)	Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.
	Fill (see page 97)	Calls the database and updates the contents of the collection if the data is out of sync with the specified parameters.
	GetEnumerator (see page 97)	Gets a System.Collections.IEnumerator for the collection.
	GetRssFeed (see page 97)	This method returns a Collection (see page 95) RSS feed.
	GetView (see page 97)	Gets the view data using a dataset.
	GetViewNames (see page 98)	Gets the view names.
	MoveNext (see page 98)	Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.
	Reset (see page 98)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 98)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.
	ToString (see page 98)	Wraps the resulting collection with the Ektron toolbars, which will only display when a user is logged in.
	Wrap (see page 98)	Overrideable method that's used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Collection.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

C#

```
public void CopyTo(Array array, int i);
```

Visual Basic

```
Public Function CopyTo(array As Array, i As Integer) As void
```

Collection.Fill Method

Calls the database and updates the contents of the collection if the data is out of sync with the specified parameters.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Collection.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

```
public IEnumerator GetEnumerator();
```

Visual Basic

```
Public Function GetEnumerator() As IEnumerator
```

Collection.GetRssFeed Method

This method returns a Collection (see page 95) RSS feed.

C#

```
public string GetRssFeed();
```

Visual Basic

```
Public Function GetRssFeed() As string
```

Returns

String

Collection.GetView Method

Gets the view data using a dataset.

C#

```
public DataSourceView GetView(string viewName);
```

Visual Basic

```
Public Function GetView(viewName As string) As DataSourceView
```

Parameters

Parameters	Description
string viewName	

Returns

DataSourceView

Collection.GetViewNames Method

Gets the view names.

C#

```
public ICollection GetViewNames();
```

Visual Basic

```
Public Function GetViewNames() As ICollection
```

Returns

View names

Collection.MoveNext Method

Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.

C#

```
public bool MoveNext();
```

Visual Basic

```
Public Function MoveNext() As bool
```

Returns

Boolean indicating if the move was within the range of the array.

Collection.Reset Method

Brings the index of the collection back to zero. Does not clear the collection.

C#

```
public void Reset();
```

Visual Basic

```
Public Function Reset() As void
```

Collection.this Indexer

Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.

C#

```
public Ektron.Cms.Controls.ContentItem.ContentRow this[int i];
```

Visual Basic

```
Public Sub this(i As Integer)
```

Collection.ToString Method

Wraps the resulting collection with the Ektron toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Collection.Wrap Method

Overrideable method that's used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

Collection Properties

The properties of the Collection class are listed here.

Public Properties

	Name	Description
	AddText (see page 100)	Allows overriding the derived controls default Add-menu button text (i.e. 'Add HTML Content (see page 822)' could be changed to 'Add Report')
	CacheInterval (see page 100)	Sets the cache interval in second(s).
	CollectionDescription (see page 100)	Returns the description of the given folder ID.
	CollectionTitle (see page 101)	Returns the name of the given folder ID.
	ContentParameter (see page 101)	Checks the QueryString for this value and replaces the collection with that content block when specified. Leave blank to always display the collection.
	Count (see page 101)	Indicates the number of elements in your Collection (see page 95).
	Current (see page 101)	Gets the current object in the collection.
	DefaultCollectionID (see page 102)	Gets or Sets the Default CollectionID for this Control.
	DynamicParameter (see page 102)	The parameter to query for a Dynamic Collection (see page 95).
	EkItems (see page 102)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 95). Holds more information than the exposed strongly typed this[i] index operator.
	EnablePaging (see page 103)	Enables paging if the total number of records exceeds the MaxResult value.
	GetHtml (see page 103)	If set to true, retrieves the content (Html body) of every content block in the collection for display.
	IncludeIcons (see page 103)	Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.
	IsSynchronized (see page 104)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 104)	Defines the way a link acts when a link is clicked. Choices are: • **_blank** - This target causes the link to always be loaded in a new blank window. This window is not named. • **_self** - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target. • **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** - This target makes the link load in... more (see page 104)
	MarkupLanguage (see page 104)	Enter the name of a markup file that manages the display of this server control.
	MaxResults (see page 104)	Maximum number of results to be returned. If set to zero, there is no limit.
	Random (see page 105)	If set to true, grabs a single content block randomly from the collection. Set to false for all data in the collection.

	SelTaxonomyID (see page 105)	Gets or Sets the Default TaxonomyID for this Collection (see page 95) Control.
	SuppressWrapperTags (see page 105)	Suppresses the output of span/div tags around the control.
	SyncRoot (see page 106)	Gets an object that's used to synchronize the collection.
	XmlDoc (see page 106)	Returns the Xslt object for the collection.

Collection.AddText Property

Allows overriding the derived controls default Add-menu button text (i.e. 'Add HTML Content (see page 822)' could be changed to 'Add Report')

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows overriding the derived controls default Add-menu button text (i.e. 'Add
HTML Content' could be changed to 'Add Report')")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Visible)
]
[Browsable(true)]
public virtual string AddText;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows overriding the derived controls default Add-menu button text (i.e. 'Add
HTML Content' could be changed to 'Add Report')")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Visible)
>
<Browsable(true)>
Public virtual Property AddText() As string
```

Collection.CacheInterval Property

Sets the cache interval in second(s).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Interval to cache this control in second(s).")]
[DefaultValue(0)]
[Personalizable(), WebBrowsable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Interval to cache this control in second(s).")>
<DefaultValue(0)>
<Personalizable(), WebBrowsable()>
Public virtual Property CacheInterval() As double
```

Collection.CollectionDescription Property

Returns the description of the given folder ID.

C#

```
public string CollectionDescription;
```

Visual Basic

```
Public Property CollectionDescription() As string
```

Collection.CollectionTitle Property

Returns the name of the given folder ID.

C#

```
public string CollectionTitle;
```

Visual Basic

```
Public Property CollectionTitle() As string
```

Collection.ContentParameter Property

Checks the QueryString for this value and replaces the collection with that content block when specified. Leave blank to always display the collection.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and replaces the collection with that
content block when specified. Leave blank to always display the collection.")]
[Personalizable(), WebBrowsable()]
public string ContentParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and replaces the collection with that
content block when specified. Leave blank to always display the collection.")>
<Personalizable(), WebBrowsable()>
Public Property ContentParameter() As string
```

Collection.Count Property

Indicates the number of elements in your Collection (see page 95).

C#

```
[Category("Data")]
[Description("Indicates the number of elements in your Collection.")]
public int Count;
```

Visual Basic

```
<Category("Data")>
<Description("Indicates the number of elements in your Collection.")>
Public Property Count() As Integer
```

Collection.Current Property

Gets the current object in the collection.

C#

```
[Browsable(false)]
public object Current;
```

Visual Basic

```
<Browsable(false)>
Public Property Current() As object
```

Collection.DefaultCollectionID Property

Gets or Sets the Default CollectionID for this Control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Editor(typeof(CollectionUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Description("Gets or Sets the Default CollectionID for this Control.")]
[Personalizable(), WebBrowsable()]
public long DefaultCollectionID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Editor(typeof(CollectionUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("Gets or Sets the Default CollectionID for this Control.")>
<Personalizable(), WebBrowsable()>
Public Property DefaultCollectionID() As long
```

Collection.DynamicParameter Property

The parameter to query for a Dynamic Collection (see page 95).

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[Description("The parameter to query for a Dynamic Collection.")]
[TypeConverter(typeof(DynamicCollectionTypeConverter))]
[Personalizable(), WebBrowsable()]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("None - Use Default")>
<Description("The parameter to query for a Dynamic Collection.")>
<TypeConverter(typeof(DynamicCollectionTypeConverter))>
<Personalizable(), WebBrowsable()>
Public Property DynamicParameter() As string
```

Collection.EkItems Property

Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 95). Holds more information than the exposed strongly typed this[i] index operator.

C#

```
[Browsable(false)]
public Ektron.Cms.Common.ContentBase EkItems;
```

Visual Basic

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

Collection.EnablePaging Property

Enables paging if the total number of records exceeds the MaxResult value.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enables paging if the total number of records exceeds the MaxResult value. Use
this property with non zero value of MaxNumbers.")]
[Personalizable(), WebBrowsable()]
public bool EnablePaging;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Enables paging if the total number of records exceeds the MaxResult value. Use
this property with non zero value of MaxNumbers.")>
<Personalizable(), WebBrowsable()>
Public Property EnablePaging() As bool
```

Collection.GetHtml Property

If set to true, retrieves the content (Html body) of every content block in the collection for display.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, retrieves the content (Html body) of every content block in the
collection for display.")]
[Personalizable(), WebBrowsable()]
public bool GetHtml;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true, retrieves the content (Html body) of every content block in the
collection for display.")>
<Personalizable(), WebBrowsable()>
Public Property GetHtml() As bool
```

Collection.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.

C#

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
[Personalizable(), WebBrowsable()]
public bool IncludeIcons;
```

Visual Basic

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")>
```

```
<DefaultValue(false)>
<Category("Ektron")>
<Personalizable(), WebBrowsable()>
Public Property IncludeIcons() As bool
```

Collection.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

```
[Browsable(false)]
public bool IsSynchronized;
```

Visual Basic

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

Collection.LinkTarget Property

Defines the way a link acts when a link is clicked. Choices are:

- **_blank** - This target causes the link to always be loaded in a new blank window. This window is not named.
- **_self** - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target.
- **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.
- **_top** - This target makes the link load in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

Collection.MarkupLanguage Property

Enter the name of a markup file that manages the display of this server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

Collection.MaxResults Property

Maximum number of results to be returned. If set to zero, there is no limit.

C#

```
[Bindable(true)]
```

```
[Category("Ektron")]
[DefaultValue(0)]
[Description("The maximum number of results to fetch. If set to zero, there is no limit.")]
[Personalizable(), WebBrowsable()]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The maximum number of results to fetch. If set to zero, there is no limit.")>
<Personalizable(), WebBrowsable()>
Public Property MaxResults() As Integer
```

Collection.Random Property

If set to true, grabs a single content block randomly from the collection. Set to false for all data in the collection.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, grabs a single content block randomly from the collection. Set
to false for all data in the collection.")]
[Personalizable(), WebBrowsable()]
public bool Random;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true, grabs a single content block randomly from the collection. Set
to false for all data in the collection.")>
<Personalizable(), WebBrowsable()>
Public Property Random() As bool
```

Collection.SelTaxonomyID Property

Gets or Sets the Default TaxonomyID for this Collection (see page 95) Control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Gets or Sets the TaxonomyID for this Collection Control.")]
[Personalizable(), WebBrowsable()]
public long SelTaxonomyID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Gets or Sets the TaxonomyID for this Collection Control.")>
<Personalizable(), WebBrowsable()>
Public Property SelTaxonomyID() As long
```

Collection.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

C#

```
public override bool SuppressWrapperTags;
```

Visual Basic

```
Public override Property SuppressWrapperTags() As bool
```

Collection.SyncRoot Property

Gets an object that's used to synchronize the collection.

C#

```
[Browsable(false)]
public object SyncRoot;
```

Visual Basic

```
<Browsable(false)>
Public Property SyncRoot() As object
```

Collection.XmlDoc Property

Returns the Xslt object for the collection.

C#

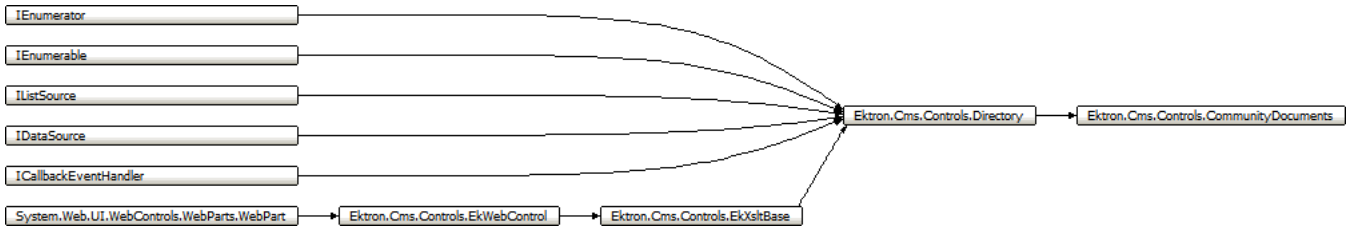
```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

CommunityDocuments Class

Class Hierarchy



C#

```
[DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:CommunityDocuments
runat=server></{0}:CommunityDocuments>")]
public class CommunityDocuments : Directory;
```

Visual Basic

```
<DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:CommunityDocuments
runat=server></{0}:CommunityDocuments>")>
Public Class CommunityDocuments
Inherits Directory
```

File

communitydocuments.cs

Description

This is class Ektron.Cms.Controls.CommunityDocuments.

CommunityDocuments Methods

The methods of the CommunityDocuments class are listed here.

Public Methods

	Name	Description
	Fill (see page 107)	Populates the data for the Taxonomy.

CommunityDocuments.Fill Method

Populates the data for the Taxonomy.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

CommunityDocuments Properties

The properties of the CommunityDocuments class are listed here.

Public Properties

	Name	Description
	AddItemFolderId (see page 107)	This is a readonly property.
	DefaultObjectID (see page 108)	Gets or Sets the default user ID for this control. This is the user ID that will be shown if the DynamicParameter (see page 202) does not get a value. (e.g. userprofile.aspx?userid=2 will get the value 2 for a DynamicParameter (see page 202) of "id", and will display DefaultUserID otherwise.) The object ID for the object that the social bar will be drawn for.
	DynamicContentBoxHeight (see page 108)	Get or sets the height of dynamic content. Value should be in pixels.
	DynamicContentBoxWidth (see page 108)	Get or sets the with of dynamic content. Value should be in pixels.
	DynamicContentTemplate (see page 109)	Get or sets the template of dynamic content. If specified control will not use dynamic box.
	EnableAjax (see page 109)	This is a readonly property.
	EnableSearch (see page 109)	This is a readonly property.
	MemberMenuActive (see page 109)	This is a readonly property.
	ObjectType (see page 110)	Select the type of object with which this control is associated.
	Recursive (see page 110)	This is a readonly property.
	ShowAllChecked (see page 110)	This is a readonly property.
	Stylesheet (see page 110)	Specify the style sheet relative or absolute. Leave blank for default.
	SuppressWrapperTags (see page 111)	This is a readonly property.
	TaxonomyDepth (see page 111)	This is a readonly property.
	TaxonomyId (see page 111)	This is a readonly property.

CommunityDocuments.AddItemFolderId Property

This is a readonly property.

C#

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override long AddItemFolderId;
```

Visual Basic

```
<Category("Ektron")>
```

```
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property AddItemFolderId() As long
```

CommunityDocuments.DefaultObjectID Property

Gets or Sets the default user ID for this control. This is the user ID that will be shown if the DynamicParameter (see page 202) does not get a value. (e.g. userprofile.aspx?userid=2 will get the value 2 for a DynamicParameter (see page 202) of "id", and will display DefaultUserID otherwise.)

The object ID for the object that the social bar will be drawn for.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Gets or sets the default object ID for this control. This is the object ID that
will be shown if the DynamicParameter property does not get a value.")]
public long DefaultObjectID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Gets or sets the default object ID for this control. This is the object ID that
will be shown if the DynamicParameter property does not get a value.")>
Public Property DefaultObjectID() As long
```

CommunityDocuments.DynamicContentBoxHeight Property

Get or sets the height of dynamic content. Value should be in pixels.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(520)]
[Description("Get or sets the height of dynamic content box. Value should be in pixels.")]
public int DynamicContentBoxHeight;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(520)>
<Description("Get or sets the height of dynamic content box. Value should be in pixels.")>
Public Property DynamicContentBoxHeight() As Integer
```

CommunityDocuments.DynamicContentBoxWidth Property

Get or sets the with of dynamic content. Value should be in pixels.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(790)]
[Description("Get or sets the width of dynamic content box. Value should be enter in pixels.")]
public int DynamicContentBoxWidth;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(790)>
<Description("Get or sets the width of dynamic content box. Value should be enter in pixels.")>
Public Property DynamicContentBoxWidth() As Integer
```

CommunityDocuments.DynamicContentTemplate Property

Get or sets the template of dynamic content. If specified control will not use dynamic box.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Get or sets the template of dynamic content. Leave blank to use dynamic box.")]
public string DynamicContentTemplate;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Get or sets the template of dynamic content. Leave blank to use dynamic box.")>
Public Property DynamicContentTemplate() As string
```

CommunityDocuments.EnableAjax Property

This is a readonly property.

C#

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override bool EnableAjax;
```

Visual Basic

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property EnableAjax() As bool
```

CommunityDocuments.EnableSearch Property

This is a readonly property.

C#

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override Boolean EnableSearch;
```

Visual Basic

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property EnableSearch() As Boolean
```

CommunityDocuments.MemberMenuActive Property

This is a readonly property.

C#

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override bool MemberMenuActive;
```

Visual Basic

```
<Category("Ektron")>
```

```
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property MemberMenuActive() As bool
```

CommunityDocuments.ObjectType Property

Select the type of object with which this control is associated.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Select the type of object with which this control is associated.")]
[DefaultValue(CommunityDocumentsObjectType.User)]
public CommunityDocumentsObjectType ObjectType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Select the type of object with which this control is associated.")>
<DefaultValue(CommunityDocumentsObjectType.User)>
Public Property ObjectType() As CommunityDocumentsObjectType
```

CommunityDocuments.Recursive Property

This is a readonly property.

C#

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override bool Recursive;
```

Visual Basic

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property Recursive() As bool
```

CommunityDocuments.ShowAllChecked Property

This is a readonly property.

C#

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override bool ShowAllChecked;
```

Visual Basic

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property ShowAllChecked() As bool
```

CommunityDocuments.Stylesheet Property

Specify the style sheet relative or absolute. Leave blank for default.

C#

```
[Description("Specify the style sheet relative or absolute. Leave blank for default."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(CssTypeConverter))]
```

```
[DefaultValue("None - Use Default")]
public string Stylesheet;
```

Visual Basic

```
<Description("Specify the style sheet relative or absolute. Leave blank for default."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(CssTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property Stylesheet() As string
```

CommunityDocuments.SuppressWrapperTags Property

This is a readonly property.

C#

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override bool SuppressWrapperTags;
```

Visual Basic

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property SuppressWrapperTags() As bool
```

CommunityDocuments.TaxonomyDepth Property

This is a readonly property.

C#

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override int TaxonomyDepth;
```

Visual Basic

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property TaxonomyDepth() As Integer
```

CommunityDocuments.TaxonomyId Property

This is a readonly property.

C#

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override long TaxonomyId;
```

Visual Basic

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property TaxonomyId() As long
```

CommunityGroupBrowser Class

Summary description for Message board

Class Hierarchy

C#

```
[DefaultProperty("TaxonomyId"), ToolboxData("<{0}:CommunityGroupBrowser
runat=server></{0}:CommunityGroupBrowser>")]
public class CommunityGroupBrowser : EkXsltWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("TaxonomyId"), ToolboxData("<{0}:CommunityGroupBrowser
runat=server></{0}:CommunityGroupBrowser>")>
Public Class CommunityGroupBrowser
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

File

CommunityGroupBrowser.cs

CommunityGroupBrowser Methods

The methods of the CommunityGroupBrowser class are listed here.

Public Methods

	Name	Description
	Fill (see page 112)	Populates the data for the CommunityGroupMembers (see page 124).
	ToString (see page 112)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 112)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

CommunityGroupBrowser.Fill Method

Populates the data for the CommunityGroupMembers (see page 124).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

CommunityGroupBrowser.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

CommunityGroupBrowser.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

CommunityGroupBrowser Properties

The properties of the CommunityGroupBrowser class are listed here.

Public Properties

	Name	Description
	AvatarHeight (see page 113)	Gets or Sets the avatar height. 0 is unlimited.
	AvatarWidth (see page 113)	Gets or Sets the avatar width. 0 is unlimited.
	CacheInterval (see page 114)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DisplayXslt (see page 114)	Select the type of review to display, or enter the path to a custom XSLT. (optional). Choices are:
	Link (see page 115)	Gets or Sets the url for the community group links.
	LinkTarget (see page 115)	Gets or Sets the target attribute for the community group links.
	MarkupLanguage (see page 115)	Enter name of a markup file that manages the display of this server control.
	MaxResults (see page 115)	The number of community groups to show per page when EnablePaging is true. If set to zero, there is no limit.
	Stylesheet (see page 116)	Specifies the Css to which this control will provide a link.
	TagTemplate (see page 116)	Gets or Sets the template-url for the group tag links.
	TagTemplateTarget (see page 116)	Gets or Sets the target (window) attribute for the group-tag links.
	TaxonomyId (see page 116)	Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.
	XmlDoc (see page 117)	Returns the Xslt object for the collection.

CommunityGroupBrowser.AvatarHeight Property

Gets or Sets the avatar height. 0 is unlimited.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") ]
public int AvatarHeight;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") >
Public Property AvatarHeight() As Integer
```

CommunityGroupBrowser.AvatarWidth Property

Gets or Sets the avatar width. 0 is unlimited.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") ]
```

```
public int AvatarWidth;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") >
Public Property AvatarWidth() As Integer
```

CommunityGroupBrowser.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

CommunityGroupBrowser.DisplayXslt Property

Select the type of review to display, or enter the path to a custom XSLT. (optional).

Choices are:

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltCommunityGroupMembersDataTypeConverter))]
[DefaultValue("ListMembersTable")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltCommunityGroupMembersDataTypeConverter))>
<DefaultValue("ListMembersTable")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

CommunityGroupBrowser.Link Property

Gets or Sets the url for the community group links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(""), Description("The community group link.
{0} is the group ID and {1} is the group name.") ]
public string Link;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(""), Description("The community group link.
{0} is the group ID and {1} is the group name.") >
Public Property Link() As string
```

CommunityGroupBrowser.LinkTarget Property

Gets or Sets the target attribute for the community group links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

CommunityGroupBrowser.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

CommunityGroupBrowser.MaxResults Property

The number of community groups to show per page when EnablePaging is true. If set to zero, there is no limit.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The number of community groups to show per page. If set to zero, there is no
limit.")]
[Personalizable()]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The number of community groups to show per page. If set to zero, there is no
limit.")>
<Personalizable()>
```

```
Public Property MaxResults() As Integer
```

CommunityGroupBrowser.Stylesheet Property

Specifies the Css to which this control will provide a link.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")>
<Personalizable()>
Public Property Stylesheet() As string
```

CommunityGroupBrowser.TagTemplate Property

Gets or Sets the template-url for the group tag links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a
community search, when a tag is clicked (example: CommunitySearch.aspx).") ]
public string TagTemplate;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a
community search, when a tag is clicked (example: CommunitySearch.aspx).") >
Public Property TagTemplate() As string
```

CommunityGroupBrowser.TagTemplateTarget Property

Gets or Sets the target (window) attribute for the group-tag links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group tag links (_blank, _self, _parent or _top), to select opening in
the same window, a popup, etc.") ]
public ItemLinkTargets TagTemplateTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group tag links (_blank, _self, _parent or _top), to select opening in
the same window, a popup, etc.") >
Public Property TagTemplateTarget() As ItemLinkTargets
```

CommunityGroupBrowser.TaxonomyId Property

Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)]
[Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")]
[Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long TaxonomyId;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)>
<Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")>
<Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property TaxonomyId() As long
```

CommunityGroupBrowser.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

CommunityGroupList Class

Summary description for Message board

Class Hierarchy

C#

```
[DefaultProperty("DefaultUserID"), ToolboxData("<{0}:CommunityGroupList
runat=server></{0}:CommunityGroupList>")]
public class CommunityGroupList : EkXsltWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultUserID"), ToolboxData("<{0}:CommunityGroupList
runat=server></{0}:CommunityGroupList>")>
Public Class CommunityGroupList
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

File

CommunityGroupList.cs

CommunityGroupList Methods

The methods of the CommunityGroupList class are listed here.

Public Methods

	Name	Description
	Fill (see page 118)	Populates the data for the CommunityGroupMembers (see page 124).
	ToString (see page 118)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 118)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

CommunityGroupList.Fill Method

Populates the data for the CommunityGroupMembers (see page 124).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

CommunityGroupList.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

CommunityGroupList.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

CommunityGroupList Properties

The properties of the CommunityGroupList class are listed here.

Public Properties

	Name	Description
	AllowGroupCreation (see page 119)	If the user has proper permissions, shows the Create Group option.
	AvatarHeight (see page 119)	Gets or Sets the avatar height. 0 is unlimited.
	AvatarWidth (see page 119)	Gets or Sets the avatar width. 0 is unlimited.
	CacheInterval (see page 119)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DefaultUserID (see page 120)	When in DisplayMode (see page 120) is "MyGroups", the ID for the user that the control will use.
	DisplayMode (see page 120)	The type of community group list you would like to see.
	DisplayXslt (see page 120)	Select the type of review to display, or enter the path to a custom XSLT. (optional). Choices are:
	DynamicUserParameter (see page 121)	When in DisplayMode (see page 120) is "MyGroups" or "MyPendingGroups", Checks the QueryString for this value and uses that value when specified. Leave blank to always use the default user ID.

	LanguageId (see page 121)	
	Link (see page 122)	Gets or Sets the url for the community group links.
	LinkTarget (see page 122)	Gets or Sets the target attribute for the community group links.
	MarkupLanguage (see page 122)	Enter name of a markup file that manages the display of this server control.
	MaxResults (see page 122)	The number of community groups to show per page when EnablePaging is true. If set to zero, there is no limit.
	ShowMySortingOptions (see page 123)	Should the My Groups, Pending and Invited Sorting Options Be Shown?
	ShowSortingOptions (see page 123)	Should the Newest, Most Popular and Group Name Sorting Options Be Shown?
	Stylesheet (see page 123)	Specifies the Css to which this control will provide a link.
	TagTemplate (see page 123)	Gets or Sets the template-url for the group tag links.
	TagTemplateTarget (see page 124)	Gets or Sets the target (window) attribute for the group-tag links.
	XmlDoc (see page 124)	Returns the Xslt object for the collection.

CommunityGroupList.AllowGroupCreation Property

If the user has proper permissions, shows the Create Group option.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(false), Description("If the user has proper
permissions, shows the Create Group option.") ]
public bool AllowGroupCreation;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(false), Description("If the user has proper
permissions, shows the Create Group option.") >
Public Property AllowGroupCreation() As bool
```

CommunityGroupList.AvatarHeight Property

Gets or Sets the avatar height. 0 is unlimited.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") ]
public int AvatarHeight;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") >
Public Property AvatarHeight() As Integer
```

CommunityGroupList.AvatarWidth Property

Gets or Sets the avatar width. 0 is unlimited.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") ]
public int AvatarWidth;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") >
Public Property AvatarWidth() As Integer
```

CommunityGroupList.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a

control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

CommunityGroupList.DefaultUserID Property

When in DisplayMode (see page 120) is "MyGroups", the ID for the user that the control will use.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("When in DisplayMode is \"MyGroups\", the ID for the user that the control will
use.")]
public long DefaultUserID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("When in DisplayMode is \"MyGroups\", the ID for the user that the control will
use.")>
Public Property DefaultUserID() As long
```

CommunityGroupList.DisplayMode Property

The type of community group list you would like to see.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Name), Description("The type of
community group list you would like to see.")]
public eDisplayMode DisplayMode;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Name), Description("The type of
community group list you would like to see.")>
Public Property DisplayMode() As eDisplayMode
```

CommunityGroupList.DisplayXslt Property

Select the type of review to display, or enter the path to a custom XSLT. (optional).

Choices are:

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltCommunityGroupMembersDataTypeConverter))]
[DefaultValue("ListMembersTable")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltCommunityGroupMembersDataTypeConverter))>
<DefaultValue("ListMembersTable")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

CommunityGroupList.DynamicUserParameter Property

When in DisplayMode (see page 120) is "MyGroups" or "MyPendingGroups", Checks the QueryString for this value and uses that value when specified. Leave blank to always use the default user ID.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("When in DisplayMode is \"MyGroups\" or \"MyPendingGroups\", checks the
QueryString for this value and uses that value when specified. Leave blank to always use the
default user ID.")]
[Personalizable()]
public string DynamicUserParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("When in DisplayMode is \"MyGroups\" or \"MyPendingGroups\", checks the
QueryString for this value and uses that value when specified. Leave blank to always use the
default user ID.")>
<Personalizable()>
Public Property DynamicUserParameter() As string
```

CommunityGroupList.LanguageId Property

C#

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("")]
[DefaultValue(0)]
public int LanguageId;
```

Visual Basic

```
<Bindable(true)>
```

```
<Browsable(false)>
<Category("Ektron")>
<Description("")>
<DefaultValue(0)>
Public Property LanguageId() As Integer
```

CommunityGroupList.Link Property

Gets or Sets the url for the community group links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(""), Description("The community group link.
{0} is the group ID and {1} is the group name.") ]
public string Link;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(""), Description("The community group link.
{0} is the group ID and {1} is the group name.") >
Public Property Link() As string
```

CommunityGroupList.LinkTarget Property

Gets or Sets the target attribute for the community group links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

CommunityGroupList.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

CommunityGroupList.MaxResults Property

The number of community groups to show per page when EnablePaging is true. If set to zero, there is no limit.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The number of community groups to show per page.If set to zero, there is no
limit.")]
[Personalizable()]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
```

```
<Category("Ektron")>
<DefaultValue(10)>
<Description("The number of community groups to show per page.If set to zero, there is no
limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

CommunityGroupList.ShowMySortingOptions Property

Should the My Groups, Pending and Invited Sorting Options Be Shown?

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true), Description("For an admin or any
user, show the My Groups, Pending and Invited sorting options.")]
public bool ShowMySortingOptions;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true), Description("For an admin or any
user, show the My Groups, Pending and Invited sorting options.")>
Public Property ShowMySortingOptions() As bool
```

CommunityGroupList.ShowSortingOptions Property

Should the Newest, Most Popular and Group Name Sorting Options Be Shown?

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true), Description("For an admin or any
user, show the Newest, Most Popular and Group Name sorting options.")]
public bool ShowSortingOptions;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true), Description("For an admin or any
user, show the Newest, Most Popular and Group Name sorting options.")>
Public Property ShowSortingOptions() As bool
```

CommunityGroupList.Stylesheet Property

Specifies the Css to which this control will provide a link.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")>
<Personalizable()>
Public Property Stylesheet() As string
```

CommunityGroupList.TagTemplate Property

Gets or Sets the template-url for the group tag links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a
community search, when a tag is clicked (example: CommunitySearch.aspx).") ]
public string TagTemplate;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a
community search, when a tag is clicked (example: CommunitySearch.aspx).") >
Public Property TagTemplate() As string
```

CommunityGroupList.TagTemplateTarget Property

Gets or Sets the target (window) attribute for the group-tag links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group tag links (_blank, _self, _parent or _top), to select opening in
the same window, a popup, etc.") ]
public ItemLinkTargets TagTemplateTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group tag links (_blank, _self, _parent or _top), to select opening in
the same window, a popup, etc.") >
Public Property TagTemplateTarget() As ItemLinkTargets
```

CommunityGroupList.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

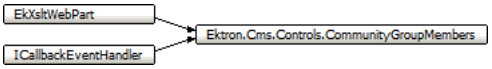
Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

CommunityGroupMembers Class

Summary description for Message board

Class Hierarchy



C#

```
[DefaultProperty("DefaultCommunityGroupID"), ToolboxData("<{0}:CommunityGroupMembers
runat=server></{0}:CommunityGroupMembers>")]
public class CommunityGroupMembers : EkXsltWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultCommunityGroupID"), ToolboxData("<{0}:CommunityGroupMembers
runat=server></{0}:CommunityGroupMembers>")>
Public Class CommunityGroupMembers
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

File

CommunityGroupMembers.cs

CommunityGroupMembers Methods

The methods of the CommunityGroupMembers class are listed here.

Public Methods

	Name	Description
	Fill (see page 125)	Populates the data for the CommunityGroupMembers (see page 124).
	ToString (see page 125)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 125)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

CommunityGroupMembers.Fill Method

Populates the data for the CommunityGroupMembers (see page 124).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

CommunityGroupMembers.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

CommunityGroupMembers.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

CommunityGroupMembers Properties

The properties of the CommunityGroupMembers class are listed here.

Public Properties

	Name	Description
	AvatarHeight (see page 126)	Gets or Sets the avatar height. 0 is unlimited.
	AvatarWidth (see page 126)	Gets or Sets the avatar width. 0 is unlimited.

	CacheInterval (see page 126)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DefaultCommunityGroupID (see page 127)	The ID for the community group that the control will use.
	DisplayMode (see page 127)	Display Mode
	DisplayXslt (see page 127)	Select the type of review to display, or enter the path to a custom XSLT. (optional). Choices are:
	DynamicParameter (see page 128)	Checks the QueryString for this value and replaces the list summary with that content block when specified. Leave blank to always display the list summary.
	Link (see page 128)	Gets or Sets the url for the user links.
	LinkTarget (see page 128)	Gets or Sets the target attribute for the user links.
	MarkupLanguage (see page 129)	Enter name of a markup file that manages the display of this server control.
	MaxResults (see page 129)	Maximum number of results to be returned.
	Stylesheet (see page 129)	Specifies the Css to which this control will provide a link.
	XmlDoc (see page 130)	Returns the Xslt object for the collection.

CommunityGroupMembers.AvatarHeight Property

Gets or Sets the avatar height. 0 is unlimited.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") ]
public int AvatarHeight;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") >
Public Property AvatarHeight() As Integer
```

CommunityGroupMembers.AvatarWidth Property

Gets or Sets the avatar width. 0 is unlimited.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") ]
public int AvatarWidth;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") >
Public Property AvatarWidth() As Integer
```

CommunityGroupMembers.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
```

```
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

CommunityGroupMembers.DefaultCommunityGroupID Property

The ID for the community group that the control will use.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID for the community group that the control will use.")]
public long DefaultCommunityGroupID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID for the community group that the control will use.")>
Public Property DefaultCommunityGroupID() As long
```

CommunityGroupMembers.DisplayMode Property

Display Mode

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Members), Description("This
property enables the user to show the community members list either with the Members or
Pending members display mode")]
public eDisplayMode DisplayMode;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Members), Description("This
property enables the user to show the community members list either with the Members or
Pending members display mode")>
Public Property DisplayMode() As eDisplayMode
```

CommunityGroupMembers.DisplayXslt Property

Select the type of review to display, or enter the path to a custom XSLT. (optional).

Choices are:

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltCommunityGroupMembersDataTypeConverter))]
```

```
[DefaultValue("ListMembersTable")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltCommunityGroupMembersDataTypeConverter))>
<DefaultValue("ListMembersTable")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

CommunityGroupMembers.DynamicParameter Property

Checks the QueryString for this value and replaces the list summary with that content block when specified. Leave blank to always display the list summary.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and uses that value when specified. Leave
blank to always use the default community group ID.")]
[Personalizable()]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and uses that value when specified. Leave
blank to always use the default community group ID.")>
<Personalizable()>
Public Property DynamicParameter() As string
```

CommunityGroupMembers.Link Property

Gets or Sets the url for the user links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(""), Description("The user link. {0} is the
user ID and {1} is the user displayname.") ]
public string Link;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(""), Description("The user link. {0} is the
user ID and {1} is the user displayname.") >
Public Property Link() As string
```

CommunityGroupMembers.LinkTarget Property

Gets or Sets the target attribute for the user links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
```

```
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

CommunityGroupMembers.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

CommunityGroupMembers.MaxResults Property

Maximum number of results to be returned.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of users to fetch per page. If set to zero, there is no
limit.")]
[Personalizable()]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("The maximum number of users to fetch per page. If set to zero, there is no
limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

CommunityGroupMembers.Stylesheet Property

Specifies the Css to which this control will provide a link.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")>
<Personalizable()>
Public Property Stylesheet() As string
```

CommunityGroupMembers.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

CommunityGroupProfile Class

Class Hierarchy

C#

```
[DefaultProperty("DefaultGroupID"), ToolboxData("<{0}:CommunityGroupProfile
runat=server></{0}:CommunityGroupProfile>")]
public class CommunityGroupProfile : EkXsltWebPart;
```

Visual Basic

```
<DefaultProperty("DefaultGroupID"), ToolboxData("<{0}:CommunityGroupProfile
runat=server></{0}:CommunityGroupProfile>")>
Public Class CommunityGroupProfile
Inherits EkXsltWebPart
```

File

Group.cs

CommunityGroupProfile Methods

The methods of the CommunityGroupProfile class are listed here.

Public Methods

	Name	Description
	Fill (see page 130)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 130)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 131)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

CommunityGroupProfile.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

CommunityGroupProfile.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

CommunityGroupProfile.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

CommunityGroupProfile Properties

The properties of the CommunityGroupProfile class are listed here.

Public Properties

	Name	Description
	DefaultGroupID (see page 131)	
	DisplayXslt (see page 132)	
	DynamicParameter (see page 132)	
	EkItem (see page 132)	
	GroupAdminLink (see page 133)	Gets or Sets the url for the group admin links.
	GroupAdminLinkTarget (see page 133)	Gets or Sets the target attribute for the group admin links.
	Stylesheet (see page 133)	Specifies the Css to which this control will provide a link.
	TagTemplate (see page 133)	Gets or Sets the template-url for the group tag links.
	TagTemplateTarget (see page 134)	Gets or Sets the target (window) attribute for the group-tag links.
	Text (see page 134)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.
	XmlDoc (see page 134)	Returns the Xslt object for the collection.

CommunityGroupProfile.DefaultGroupID Property

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The ID for the group that is used")]
[TypeConverter(typeof(GroupUITypeEditor))]
[DefaultValue(0)]
public long DefaultGroupID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The ID for the group that is used")>
<TypeConverter(typeof(GroupUITypeEditor))>
```

```
<DefaultValue(0)>
Public Property DefaultGroupID() As long
```

CommunityGroupProfile.DisplayXslt Property

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltGroupTypeConverter))]
[DefaultValue("Default")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltGroupTypeConverter))>
<DefaultValue("Default")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

CommunityGroupProfile.DynamicParameter Property

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("To make this group control dynamic, select id. When you do, this server control
is attached to the community group passed as a URL parameter.")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("To make this group control dynamic, select id. When you do, this server control
is attached to the community group passed as a URL parameter.")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

CommunityGroupProfile.EkItem Property

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The CommunityGroupData Object.")]
[TypeConverter(typeof(GroupUITypeEditor))]
[Browsable(false)]
public CommunityGroupData EkItem;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The CommunityGroupData Object.")>
<TypeConverter(typeof(GroupUITypeEditor))>
<Browsable(false)>
Public Property EkItem() As CommunityGroupData
```

CommunityGroupProfile.GroupAdminLink Property

Gets or Sets the url for the group admin links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue("Use QuickLink"),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The group admin link. {0} is the
group admin ID and {1} is the group admin displayname.") ]
public string GroupAdminLink;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue("Use QuickLink"),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The group admin link. {0} is the
group admin ID and {1} is the group admin displayname.") >
Public Property GroupAdminLink() As string
```

CommunityGroupProfile.GroupAdminLinkTarget Property

Gets or Sets the target attribute for the group admin links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group admin links.") ]
public ItemLinkTargets GroupAdminLinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group admin links.") >
Public Property GroupAdminLinkTarget() As ItemLinkTargets
```

CommunityGroupProfile.Stylesheet Property

Specifies the Css to which this control will provide a link.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")>
<Personalizable()>
Public Property Stylesheet() As string
```

CommunityGroupProfile.TagTemplate Property

Gets or Sets the template-url for the group tag links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a
community search, when a tag is clicked (example: CommunitySearch.aspx).") ]
```

```
public string TagTemplate;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a
community search, when a tag is clicked (example: CommunitySearch.aspx).") >
Public Property TagTemplate() As string
```

CommunityGroupProfile.TagTemplateTarget Property

Gets or Sets the target (window) attribute for the group-tag links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group tag links (_blank, _self, _parent or _top), to select opening in
the same window, a popup, etc.") ]
public ItemLinkTargets TagTemplateTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group tag links (_blank, _self, _parent or _top), to select opening in
the same window, a popup, etc.") >
Public Property TagTemplateTarget() As ItemLinkTargets
```

CommunityGroupProfile.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to treat the server control as a Label, taking information
from the Control and painting it on top of the Control for display. Leave blank for default
display.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public new virtual string Text;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows a developer to treat the server control as a Label, taking information
from the Control and painting it on top of the Control for display. Leave blank for default
display.")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Browsable(false)>
Public new virtual Property Text() As string
```

CommunityGroupProfile.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

CommunitySearch Class

Class Hierarchy

C#

```
[DefaultProperty("DefaultCommunitySearchID"), ToolboxData("<{0}:CommunitySearch
runat=server></{0}:CommunitySearch>")]
public class CommunitySearch : EkXsltWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultCommunitySearchID"), ToolboxData("<{0}:CommunitySearch
runat=server></{0}:CommunitySearch>")>
Public Class CommunitySearch
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

File

CommunitySearch.cs

Description

This is class Ektron.Cms.Controls.CommunitySearch.

CommunitySearch Methods

The methods of the CommunitySearch class are listed here.

Public Methods

	Name	Description
	Fill (see page 135)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 135)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 136)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

CommunitySearch.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

CommunitySearch.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

CommunitySearch.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

CommunitySearch Properties

The properties of the CommunitySearch class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 137)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300.
	CustomPropertyID (see page 138)	ID (or comma seperated IDs) of the Custom Property to render in the information box.
	DisplayXslt (see page 138)	The Xslt to use to render the Control (optional)
	EnableFilteringResults (see page 139)	If true, matching taxonomy categories are shown, allowing the user to narrow the resuls.
	EnableGroupResults (see page 139)	If false, groups are prevented from being returned in the results.
	EnableMap (see page 139)	Set to false to hide the map functionality in this control. This hides the Location tab when the control is displayed on a Web site and the map properties in Visual Studio.
	EnableMouseOutHidesInfo (see page 140)	When true, mousing away from a users name hides the pop-up info box.
	EnableMouseOver (see page 140)	When set to False, the profile pop-up is disabled when hovering over a display name in the search results. • True = Enable the profile pop-up when hovering over a display name. • False = Disable the profile pop-up when hovering over a display name.
	EnableUnauthorizedSearching (see page 140)	When set to True, a user does not need to be logged in to perform a user search. When set to False, a user must logged in to perform a user search, otherwise the control displays a message stating that log-in is required. • **True** = a user does not need to be logged in to perform a user search. • **False** = a user must be logged in to perform a user search.
	EnableUserResults (see page 141)	If false, users are prevented from being returned in the results.
	FriendsOnly (see page 141)	If true, only returns friends.
	GroupTaxonomyID (see page 141)	The taxonomy ID to use for the group-categories (if zero, then none shown). This category must be created through the taxonomy wizard.
	InfoBoxShowsEmail (see page 142)	Set true to include users email address values in the pop-up information box (visible by mousing over a users name in the search results).
	LoadingImage (see page 142)	Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**.
	MapControl (see page 142)	This property allows you to set properties for the Location Tab's map. These properties are identical to the properties in the Map (see page 350) server control. The EnableMap (see page 139) property must be set to true to see the sub properties.

	MarkupLanguage (see page 143)	Enter the name of an EkML file that manages the display of this server control.
	MaxTagCount (see page 143)	The maximum number of personal tags to show for each user in the search results.
	MembersOnly (see page 143)	When set to True, this server control only returns Membership (see page 360) Users (see page 1076). When set to False, the control returns CMS400.NET Users (see page 1076) and Membership (see page 360) Users (see page 1076).
	PageSize (see page 144)	The maximum number of user results to show per page.
	ResultProfile (see page 144)	Determines whether profile information is displayed. This is based on the state of the user performing the user search. Choices are: • **full** - always show user profile information. • **nonprivate** - display user profile information only when a user's profile is not marked private. • **nonprivate-authenticated** - display user profile information only when a user's profile is not marked private and the user performing the search is logged in. • **none** - never show user profile information.
	SearchPrivate (see page 144)	When set to False, this property prevents users with profiles marked as private from being returned with user search results.
	ShowNameEmail (see page 145)	When set to True, populates the drop down list with these additional options: • **First Name** - perform searches based on a user's first name. • **Last Name** - perform searches based on a user's last name. • **Email** - perform searches based on a user's email address. When set to False, the drop down list is populated with these properties: • **Tags (see page 805)** - perform searches based on a user's tags. • **Display Name** - perform searches based on a user's display name.
	StartingTab (see page 145)	Default tab that's selected when the server control is first rendered on the Web site. Choose from **basic**, **advanced** or **location**.
	Stylesheet (see page 146)	Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.
	TemplateGroupProfile (see page 146)	The URL path of the community group template. This path can be relative or absolute.
	TemplateParamName (see page 146)	Gets or sets a Querystring parameter for the ID of users or groups that are returned in the search results. This parameter is passed to the template page defined in the TemplateUserProfile (see page 147) or TemplateGroupProfile (see page 146) property when a user clicks the Location (map) icon associated with a search result.
	TemplateTarget (see page 147)	Determines the type of window that appears on this Web form when a user clicks an a user name or tag. The default is _self. • **_self** = opens in same window • **_top** = opens in parent window • **_blank** = opens in new window • **_parent** = opens in the parent frame
	TemplateUserProfile (see page 147)	The URL path of the user profile template. This path can be relative or absolute.
	UseMsAjax (see page 147)	When true, uses Microsoft Ajax.
	UserTaxonomyID (see page 148)	The taxonomy ID to use for the user-categories (if zero, then none shown). This category must be created through the taxonomy wizard.
	XmlDoc (see page 148)	Returns the Xml object for the control.
	XsltDoc (see page 148)	Returns the Xsl object for the control.

CommunitySearch.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a

control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(double))]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue("0")]
public double CacheInterval;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(double))>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue("0")>
Public Property CacheInterval() As double
```

CommunitySearch.CustomPropertyID Property

ID (or comma seperated IDs) of the Custom Property to render in the information box.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("Specifies the custom properties to include in the users information pop-up box,
after the first and last names, in the search results (use a comma-delimiter for more than one
ID)")]
[DefaultValue("")]
public string CustomPropertyID;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("Specifies the custom properties to include in the users information pop-up box,
after the first and last names, in the search results (use a comma-delimiter for more than one
ID)")>
<DefaultValue("")>
Public Property CustomPropertyID() As string
```

CommunitySearch.DisplayXslt Property

The Xslt to use to render the Control (optional)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltCommunitySearchTypeConverter))]
[DefaultValue("CommunitySearchDirectory.xsl")]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltCommunitySearchTypeConverter))>
<DefaultValue("CommunitySearchDirectory.xsl")>
Public override Property DisplayXslt() As string
```

CommunitySearch.EnableFilteringResults Property

If true, matching taxonomy categories are shown, allowing the user to narrow the resuls.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Set to true to enable showing any matching taxonomy categories, which the user
can then use to reduce the results to suit their interests")]
[DefaultValue("true")]
public bool EnableFilteringResults;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Set to true to enable showing any matching taxonomy categories, which the user
can then use to reduce the results to suit their interests")>
<DefaultValue("true")>
Public Property EnableFilteringResults() As bool
```

CommunitySearch.EnableGroupResults Property

If false, groups are prevented from being returned in the results.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Set to true to enable groups being included in the results (default), false
prevents groups from being returned in the results")]
[DefaultValue("true")]
public bool EnableGroupResults;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Set to true to enable groups being included in the results (default), false
prevents groups from being returned in the results")>
<DefaultValue("true")>
Public Property EnableGroupResults() As bool
```

CommunitySearch.EnableMap Property

Set to false to hide the map functionality in this control. This hides the Location tab when the control is displayed on a Web site and the map properties in Visual Studio.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Set to false to hide the map functionality in this control. This hides the
Location tab when the control is displayed on a Web site and the map properties in Visual
Studio.")]
[DefaultValue("true")]
public bool EnableMap;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Set to false to hide the map functionality in this control. This hides the
Location tab when the control is displayed on a Web site and the map properties in Visual
Studio.")>
<DefaultValue("true")>
Public Property EnableMap() As bool
```

CommunitySearch.EnableMouseOutHidesInfo Property

When true, mousing away from a users name hides the pop-up info box.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("If true, then the the pop-up information box is hidden when the mouse moves away
from a users display name in the search results. If false, then the close button must be
clicked.")]
[DefaultValue("true")]
public bool EnableMouseOutHidesInfo;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("If true, then the the pop-up information box is hidden when the mouse moves away
from a users display name in the search results. If false, then the close button must be
clicked.")>
<DefaultValue("true")>
Public Property EnableMouseOutHidesInfo() As bool
```

CommunitySearch.EnableMouseOver Property

When set to False, the profile pop-up is disabled when hovering over a display name in the search results.

- True = Enable the profile pop-up when hovering over a display name.
- False = Disable the profile pop-up when hovering over a display name.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("When the property is set to false, prevents pop-up profile from ever being
shown.")]
[DefaultValue("true")]
public bool EnableMouseOver;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("When the property is set to false, prevents pop-up profile from ever being
shown.")>
<DefaultValue("true")>
Public Property EnableMouseOver() As bool
```

CommunitySearch.EnableUnauthorizedSearching Property

When set to True, a user does not need to be logged in to perform a user search. When set to False, a user must logged in to perform a user search, otherwise the control displays a message stating that log-in is required.

- **True** = a user does not need to be logged in to perform a user search.
- **False** = a user must be logged in to perform a user search.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description(" When the property is set to false and user is not logged in, UI displays
message that log-in is required (searching is prevented).")]
[DefaultValue("true")]
public bool EnableUnauthorizedSearching;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description(" When the property is set to false and user is not logged in, UI displays
message that log-in is required (searching is prevented).")>
<DefaultValue("true")>
Public Property EnableUnauthorizedSearching() As bool
```

CommunitySearch.EnableUserResults Property

If false, users are prevented from being returned in the results.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Set to true to enable users being included in the results (default), false
prevents users from being returned in the results")]
[DefaultValue("true")]
public bool EnableUserResults;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Set to true to enable users being included in the results (default), false
prevents users from being returned in the results")>
<DefaultValue("true")>
Public Property EnableUserResults() As bool
```

CommunitySearch.FriendsOnly Property

If true, only returns friends.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Set to true to limit results to the users friends, otherwise returns all matches
(default)")]
[DefaultValue("false")]
public bool FriendsOnly;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Set to true to limit results to the users friends, otherwise returns all matches
(default)")>
<DefaultValue("false")>
Public Property FriendsOnly() As bool
```

CommunitySearch.GroupTaxonomyID Property

The taxonomy ID to use for the group-categories (if zero, then none shown). This category must be created through the taxonomy wizard.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("The taxonomy ID to use for the group-categories (if zero, then none shown). This
category must be created through the taxonomy wizard.")]
[DefaultValue("0")]
public long GroupTaxonomyID;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("The taxonomy ID to use for the group-categories (if zero, then none shown). This
category must be created through the taxonomy wizard.")>
<DefaultValue("0")>
Public Property GroupTaxonomyID() As long
```

CommunitySearch.InfoBoxShowsEmail Property

Set true to include users email address values in the pop-up information box (visible by mousing over a users name in the search results).

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("If true, then the the users email address will be rendered in the information
box (if enabled and the viewer has the required authority).")]
[DefaultValue("false")]
public bool InfoBoxShowsEmail;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("If true, then the the users email address will be rendered in the information
box (if enabled and the viewer has the required authority).")>
<DefaultValue("false")>
Public Property InfoBoxShowsEmail() As bool
```

CommunitySearch.LoadingImage Property

Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")]
[DefaultValue("ajax-loader_circle_lg.gif")]
public string LoadingImage;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")>
<DefaultValue("ajax-loader_circle_lg.gif")>
Public Property LoadingImage() As string
```

CommunitySearch.MapControl Property

This property allows you to set properties for the Location Tab's map. These properties are identical to the properties in the Map (see page 350) server control. The EnableMap (see page 139) property must be set to true to see the sub properties.

C#

```
[Browsable(false), Category("Ektron")]
[TypeConverter(typeof(Ektron.Cms.Controls.Map))]
[Description("This property allows you to set properties for the Location Tab?s map. These
properties are identical to the properties in the Map server control. The EnableMap property
must be set to true to see the sub properties.")]
public Ektron.Cms.Controls.Map MapControl;
```

Visual Basic

```
<Browsable(false), Category("Ektron")>
<TypeConverter(typeof(Ektron.Cms.Controls.Map))>
<Description("This property allows you to set properties for the Location Tab?s map. These
properties are identical to the properties in the Map server control. The EnableMap property
must be set to true to see the sub properties.")>
Public Property MapControl() As Ektron.Cms.Controls.Map
```

Description

The map control properties can be changed only through code behind and not from designer property grid

CommunitySearch.MarkupLanguage Property

Enter the name of an EkML file that manages the display of this server control.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("Enter the name of an EkML file that manages the display of this server
control.")]
[DefaultValue("")]
public string MarkupLanguage;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("Enter the name of an EkML file that manages the display of this server
control.")>
<DefaultValue("")>
Public Property MarkupLanguage() As string
```

CommunitySearch.MaxTagCount Property

The maximum number of personal tags to show for each user in the search results.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("The maximum number of personal tags to show for each user in the search
results.")]
[DefaultValue("5")]
public int MaxTagCount;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("The maximum number of personal tags to show for each user in the search
results.")>
<DefaultValue("5")>
Public Property MaxTagCount() As Integer
```

CommunitySearch.MembersOnly Property

When set to True, this server control only returns Membership (see page 360) Users (see page 1076). When set to False, the control returns CMS400.NET Users (see page 1076) and Membership (see page 360) Users (see page 1076).

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Allows limiting results for security; When true, search only returns members.
When false, returns members and CMS uers.")]
```

```
[DefaultValue("false")]
public bool MembersOnly;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Allows limiting results for security; When true, search only returns members.
When false, returns members and CMS uers.")>
<DefaultValue("false")>
Public Property MembersOnly() As bool
```

CommunitySearch.PageSize Property

The maximum number of user results to show per page.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("The maximum number of user results to show per page.")]
[DefaultValue("10")]
public int PageSize;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("The maximum number of user results to show per page.")>
<DefaultValue("10")>
Public Property PageSize() As Integer
```

CommunitySearch.ResultProfile Property

Determines whether profile information is displayed. This is based on the state of the user performing the user search. Choices are:

- **full** - always show user profile information.
- **nonprivate** - display user profile information only when a user's profile is not marked private.
- **nonprivate-authenticated** - display user profile information only when a user's profile is not marked private and the user performing the search is logged in.
- **none** - never show user profile information.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(ResultProfileCommunitySearchTypeConverter))]
[Description(" This property will display the information of users or friends based on the
state of the user who is seeking the information.")]
[DefaultValue("nonprivate-authenticated")]
public string ResultProfile;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(ResultProfileCommunitySearchTypeConverter))>
<Description(" This property will display the information of users or friends based on the
state of the user who is seeking the information.")>
<DefaultValue("nonprivate-authenticated")>
Public Property ResultProfile() As string
```

CommunitySearch.SearchPrivate Property

When set to False, this property prevents users with profiles marked as private from being returned with user search results.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("When set to False, this property prevents users with profiles marked as private
from being returned with user search results.")]
[DefaultValue("true")]
public bool SearchPrivate;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("When set to False, this property prevents users with profiles marked as private
from being returned with user search results.")>
<DefaultValue("true")>
Public Property SearchPrivate() As bool
```

CommunitySearch.ShowNameEmail Property

When set to True, populates the drop down list with these additional options:

- **First Name** - perform searches based on a user's first name.
- **Last Name** - perform searches based on a user's last name.
- **Email** - perform searches based on a user's email address.

When set to False, the drop down list is populated with these properties:

- **Tags (see page 805)** - perform searches based on a user's tags.
- **Display Name** - perform searches based on a user's display name.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("When the property is set to true, populates the drop-down list with additional
options; First Name, Last Name, email.")]
[DefaultValue("true")]
public bool ShowNameEmail;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("When the property is set to true, populates the drop-down list with additional
options; First Name, Last Name, email.")>
<DefaultValue("true")>
Public Property ShowNameEmail() As bool
```

CommunitySearch.StartingTab Property

Default tab that's selected when the server control is first rendered on the Web site. Choose from **basic**, **advanced** or **location**.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The tab to show as selected, when the control is first rendered: choose from
'basic', 'advanced' or 'location'.")]
[DefaultValue("basic")]
public string StartingTab;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The tab to show as selected, when the control is first rendered: choose from
'basic', 'advanced' or 'location'.")>
```

```
<DefaultValue("basic")>
Public Property StartingTab() As string
```

CommunitySearch.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Category("Ektron")]
[Description("Allows selecting a different CSS file for the community search server control.")]
[TypeConverter(typeof(string))]
[DefaultValue("csslib/CommunitySearch.css")]
public string Stylesheet;
```

Visual Basic

```
<Category("Ektron")>
<Description("Allows selecting a different CSS file for the community search server control.")>
<TypeConverter(typeof(string))>
<DefaultValue("csslib/CommunitySearch.css")>
Public Property Stylesheet() As string
```

CommunitySearch.TemplateGroupProfile Property

The URL path of the community group template. This path can be relative or absolute.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The URL path of the community group template. This path can be relative or
absolute.")]
[DefaultValue("")]
public string TemplateGroupProfile;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The URL path of the community group template. This path can be relative or
absolute.")>
<DefaultValue("")>
Public Property TemplateGroupProfile() As string
```

CommunitySearch.TemplateParamName Property

Gets or sets a Querystring parameter for the ID of users or groups that are returned in the search results. This parameter is passed to the template page defined in the TemplateUserProfile (see page 147) or TemplateGroupProfile (see page 146) property when a user clicks the Location (map) icon associated with a search result.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("Gets or sets a Querystring parameter for the ID of users or groups that are
returned in the search results. This parameter is passed to the template page defined in the
TemplateUserProfile or TemplateGroupProfile property when a user clicks the Location (map)
icon associated with a search result.")]
[DefaultValue("id")]
public string TemplateParamName;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("Gets or sets a Querystring parameter for the ID of users or groups that are
returned in the search results. This parameter is passed to the template page defined in the
```

```
TemplateUserProfile or TemplateGroupProfile property when a user clicks the Location (map)
icon associated with a search result.")>
<DefaultValue("id")>
Public Property TemplateParamName() As string
```

CommunitySearch.TemplateTarget Property

Determines the type of window that appears on this Web form when a user clicks an a user name or tag. The default is _self.

- **_self** = opens in same window
- **_top** = opens in parent window
- **_blank** = opens in new window
- **_parent** = opens in the parent frame

C#

```
[Category("Ektron")]
[DefaultValue(ItemLinkTargets._self)]
[Description("The target attribute of the personal-tag and profile links (_blank, _self,
_parent or _top), to select opening in the same window, a popup, etc.")]
public ItemLinkTargets TemplateTarget;
```

Visual Basic

```
<Category("Ektron")>
<DefaultValue(ItemLinkTargets._self)>
<Description("The target attribute of the personal-tag and profile links (_blank, _self,
_parent or _top), to select opening in the same window, a popup, etc.")>
Public Property TemplateTarget() As ItemLinkTargets
```

CommunitySearch.TemplateUserProfile Property

The URL path of the user profile template. This path can be relative or absolute.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The URL path of the user profile template. This path can be relative or
absolute.")]
[DefaultValue("")]
public string TemplateUserProfile;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The URL path of the user profile template. This path can be relative or
absolute.")>
<DefaultValue("")>
Public Property TemplateUserProfile() As string
```

CommunitySearch.UseMsAjax Property

When true, uses Microsoft Ajax.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("If true, uses Microsoft Ajax to perform callback. Otherwise uses Ektron Library
callback (default).")]
[DefaultValue("false")]
public bool UseMsAjax;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("If true, uses Microsoft Ajax to perform callback. Otherwise uses Ektron Library
callback (default).")>
<DefaultValue("false")>
Public Property UseMsAjax() As bool
```

CommunitySearch.UserTaxonomyID Property

The taxonomy ID to use for the user-categories (if zero, then none shown). This category must be created through the taxonomy wizard.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("The taxonomy ID to use for the user-categories (if zero, then none shown). This
category must be created through the taxonomy wizard.")]
[DefaultValue("0")]
public long UserTaxonomyID;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("The taxonomy ID to use for the user-categories (if zero, then none shown). This
category must be created through the taxonomy wizard.")>
<DefaultValue("0")>
Public Property UserTaxonomyID() As long
```

CommunitySearch.XmlDoc Property

Returns the Xml object for the control.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

CommunitySearch.XsltDoc Property

Returns the Xsl object for the control.

C#

```
public override XmlDocument XsltDoc;
```

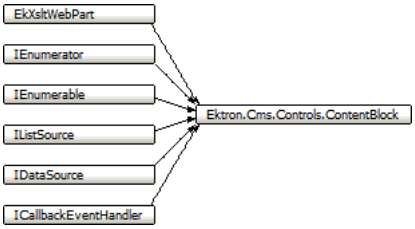
Visual Basic

```
Public override Property XsltDoc() As XmlDocument
```

ContentBlock Class

Summary description for ContentBlock.

Class Hierarchy



C#

```
[DefaultProperty("DefaultContentID"), ToolboxData("<{0}:ContentBlock
runat=server></{0}:ContentBlock>")]
public class ContentBlock : EkXsltWebPart, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultContentID"), ToolboxData("<{0}:ContentBlock
runat=server></{0}:ContentBlock>")>
Public Class ContentBlock
Inherits EkXsltWebPart
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
Inherits ICallbackEventHandler
```

File

ContentBlock.cs

ContentBlock Methods

The methods of the ContentBlock class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 149)	Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.
	Fill (see page 150)	Method that can be overridden and used to grab data from the back end for each server control.
	GetEnumerator (see page 150)	Gets a System.Collections.IEnumerator for the collection.
	MoveNext (see page 150)	Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.
	Reset (see page 150)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 150)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.
	ToString (see page 150)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 151)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

ContentBlock.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

C#

```
public void CopyTo(Array array, int i);
```

Visual Basic

```
Public Function CopyTo(array As Array, i As Integer) As void
```

ContentBlock.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

ContentBlock.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

```
public IEnumerator GetEnumerator();
```

Visual Basic

```
Public Function GetEnumerator() As IEnumerator
```

ContentBlock.MoveNext Method

Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.

C#

```
public bool MoveNext();
```

Visual Basic

```
Public Function MoveNext() As bool
```

Returns

Boolean indicating if the move was within the range of the array.

ContentBlock.Reset Method

Brings the index of the collection back to zero. Does not clear the collection.

C#

```
public void Reset();
```

Visual Basic

```
Public Function Reset() As void
```

ContentBlock.this Indexer

Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.

C#

```
public Ektron.Cms.Controls.ContentItem.ContentRow this[int i];
```

Visual Basic

```
Public Sub this(i As Integer)
```

ContentBlock.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

ContentBlock.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

Html string wrapped with a toolbar

ContentBlock Properties

The properties of the ContentBlock class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 151)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	Count (see page 152)	Indicates the number of elements in your Collection (see page 95).
	Current (see page 152)	Gets the current object in the collection.
	DefaultContentID (see page 152)	Gets or Sets the Default ContentID for this Control.
	DisplayXslt (see page 153)	Gets or Sets the path to an Xslt that can be applied to this content block. Set to String.Empty if you wish to always display the default Content (see page 822) view.
	DynamicParameter (see page 153)	Gets or Sets the QueryString parameter to read a Content (see page 822) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)
	EkItem (see page 153)	Gets the Ektron.Cms.Common.ContentBase for the current Content (see page 822) Block.
	IsSynchronized (see page 154)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	OverrideXslt (see page 154)	Ektron CMS400.NET allows an Administrator to specify up to three custom Xslt's which can be applied to a content block from the System. This allows the user to override the default ContentBlock (see page 148) Xslt and display one of these custom Xsl transforms, if specified in the system. In addition, you can use the Xslt property to perform a transform on a content block (see Xslt property.)
	StartingFolderID (see page 154)	For use with Web Parts only. Allows a developer to set the starting folder for the personalization/collaboration selection dialog.
	SyncRoot (see page 155)	Gets an object that can be used to synchronize the collection.

ContentBlock.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
```

```
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable(), WebBrowsable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable(), WebBrowsable()>
Public virtual Property CacheInterval() As double
```

ContentBlock.Count Property

Indicates the number of elements in your Collection (see page 95).

C#

```
[Browsable(false)]
public int Count;
```

Visual Basic

```
<Browsable(false)>
Public Property Count() As Integer
```

ContentBlock.Current Property

Gets the current object in the collection.

C#

```
[Browsable(false)]
public object Current;
```

Visual Basic

```
<Browsable(false)>
Public Property Current() As object
```

ContentBlock.DefaultContentID Property

Gets or Sets the Default ContentID for this Control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the Default ContentID for this Control.")]
[Personalizable(), WebBrowsable()]
public long DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the Default ContentID for this Control.")>
```

```
<Personalizable(), WebBrowsable()>
Public Property DefaultContentID() As long
```

ContentBlock.DisplayXslt Property

Gets or Sets the path to an Xslt that can be applied to this content block. Set to String.Empty if you wish to always display the default Content (see page 822) view.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter("DynamicParameterTypeConverter")]
[Description("Use this to specify an XSLT that will override the default display for an XML
content block.")]
[Personalizable(), WebBrowsable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter("DynamicParameterTypeConverter")>
<Description("Use this to specify an XSLT that will override the default display for an XML
content block.")>
<Personalizable(), WebBrowsable()>
Public override Property DisplayXslt() As string
```

ContentBlock.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 822) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Content ID dynamically. Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
[Personalizable(), WebBrowsable()]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Content ID dynamically. Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
<Personalizable(), WebBrowsable()>
Public Property DynamicParameter() As string
```

ContentBlock.EkItem Property

Gets the Ektron.Cms.Common.ContentBase for the current Content (see page 822) Block.

C#

```
[Browsable(false)]
public Ektron.Cms.Common.ContentBase EkItem;
```

Visual Basic

```
<Browsable(false)>
```

```
Public Property EkItem() As Ektron.Cms.Common.ContentBase
```

ContentBlock.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

```
[Browsable(false)]
public bool IsSynchronized;
```

Visual Basic

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

ContentBlock.OverrideXslt Property

Ektron CMS400.NET allows an Administrator to specify up to three custom Xslt's which can be applied to a content block from the System. This allows the user to override the default ContentBlock (see page 148) Xslt and display one of these custom Xsl transforms, if specified in the system. In addition, you can use the Xslt property to perform a transform on a content block (see Xslt property.)

C#

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[DefaultValue(XsltOverride.Default)]
[Description("Use this to specify an XSLT that will override the default display for an XML
content block.")]
[Obsolete("Specify the display XSLT path in \"DisplayXslt\" property.")]
public XsltOverride OverrideXslt;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<DefaultValue(XsltOverride.Default)>
<Description("Use this to specify an XSLT that will override the default display for an XML
content block.")>
<Obsolete("Specify the display XSLT path in \"DisplayXslt\" property.")>
Public Property OverrideXslt() As XsltOverride
```

ContentBlock.StartingFolderID Property

For use with Web Parts only. Allows a developer to set the starting folder for the personalization/collaboration selection dialog.

C#

```
[Bindable(true)]
[Category("Ektron WebParts")]
[Description("WEB PARTS ONLY: Allows a developer to set the starting folder for the
personalization/collaboration selection dialog.")]
[DefaultValue(0)]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long StartingFolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron WebParts")>
<Description("WEB PARTS ONLY: Allows a developer to set the starting folder for the
personalization/collaboration selection dialog.")>
<DefaultValue(0)>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
```

```
Public Property StartingFolderID() As long
```

ContentBlock.SyncRoot Property

Gets an object that can be used to synchronize the collection.

C#

```
[Browsable(false)]
public object SyncRoot;
```

Visual Basic

```
<Browsable(false)>
Public Property SyncRoot() As object
```

ContentFlagging Class

Class Hierarchy

C#

```
[DefaultProperty("DefaultContentID"), ToolboxData("<{0}:ContentFlagging
runat=server></{0}:ContentFlagging>")]
public class ContentFlagging : EkXsltWebPart;
```

Visual Basic

```
<DefaultProperty("DefaultContentID"), ToolboxData("<{0}:ContentFlagging
runat=server></{0}:ContentFlagging>")>
Public Class ContentFlagging
Inherits EkXsltWebPart
```

File

ContentFlagging.cs

ContentFlagging Methods

The methods of the ContentFlagging class are listed here.

Public Methods

	Name	Description
	Fill (see page 155)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 155)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 156)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

ContentFlagging.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

ContentFlagging.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

ContentFlagging.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

ContentFlagging Properties

The properties of the ContentFlagging class are listed here.

Public Properties

	Name	Description
	DefaultContentID (see page 156)	Gets or Sets the default content ID for this control. This is the content ID that will be flagged if the DynamicParameter (see page 157) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 157) of "id", and will display DefaultContentID otherwise.)
	DisplayXslt (see page 157)	Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.
	DynamicParameter (see page 157)	Gets or sets the Querystring parameter to read a content ID dynamically. To use the default content ID, leave blank.
	FlagImage (see page 157)	Determines the image that will be shown to allow flagging. This overrides the FlagText (see page 158) value.
	FlagText (see page 158)	Determines the text that will be shown to allow flagging. FlagImage (see page 157) overrides this value.
	Stylesheet (see page 158)	Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Select **Use Default** to use the default style sheet.
	XmlDoc (see page 159)	Returns the Xslt object for the collection.

ContentFlagging.DefaultContentID Property

Gets or Sets the default content ID for this control. This is the content ID that will be flagged if the DynamicParameter (see page 157) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 157) of "id", and will display DefaultContentID otherwise.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentFlaggingUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or sets the default content ID for this control. This is the content ID
that will be flagged if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will
get the value 2 for a DynamicParameter of \"id\", and will use DefaultContentID otherwise.)")]
public long DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentFlaggingUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or sets the default content ID for this control. This is the content ID
that will be flagged if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will
get the value 2 for a DynamicParameter of \"id\", and will use DefaultContentID otherwise.)")>
Public Property DefaultContentID() As long
```

ContentFlagging.DisplayXslt Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute.")]
[TypeConverter(typeof(XsltContentFlaggingTypeConverter))]
[DefaultValue("Dropdown with Comments")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute.")>
<TypeConverter(typeof(XsltContentFlaggingTypeConverter))>
<DefaultValue("Dropdown with Comments")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

ContentFlagging.DynamicParameter Property

Gets or sets the Querystring parameter to read a content ID dynamically. To use the default content ID, leave blank.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or sets the Querystring parameter to read a content ID dynamically. To use
the default content ID, leave blank.")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or sets the Querystring parameter to read a content ID dynamically. To use
the default content ID, leave blank.")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

ContentFlagging.FlagImage Property

Determines the image that will be shown to allow flagging. This overrides the FlagText (see page 158) value.

C#

```
[Bindable(true)]
[Category("Ektron")]
```

```
[Description("Determines the image that will be shown to allow flagging. This overrides the
FlagText value.")]
[DefaultValue("")]
[Personalizable()]
public string FlagImage;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Determines the image that will be shown to allow flagging. This overrides the
FlagText value.")>
<DefaultValue("")>
<Personalizable()>
Public Property FlagImage() As string
```

ContentFlagging.FlagText Property

Determines the text that will be shown to allow flagging. FlagImage (see page 157) overrides this value.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Determines the text that will be shown to allow flagging. FlagImage overrides
this value.")]
[DefaultValue("Flag")]
[Personalizable()]
public string FlagText;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Determines the text that will be shown to allow flagging. FlagImage overrides
this value.")>
<DefaultValue("Flag")>
<Personalizable()>
Public Property FlagText() As string
```

ContentFlagging.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Select **Use Default** to use the default style sheet.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Select Use Default to use the default style sheet.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Select Use Default to use the default style sheet.")>
<Personalizable()>
Public Property Stylesheet() As string
```

ContentFlagging.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

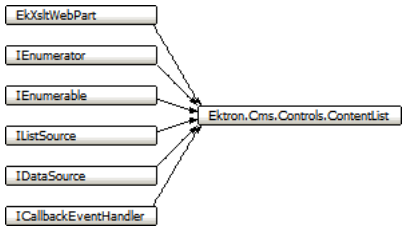
Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

ContentList Class

The ContentList server control is used to display a list of content blocks on a Web page. In contrast to a List Summary, where content must be in a specified folder, the ContentList server control uses content from any CMS400.NET folder.

Class Hierarchy



C#

```
[DefaultProperty("ContentIds"), ToolboxData("<{0}:ContentList
runat=server></{0}:ContentList>")]
public class ContentList : EkXsltWebPart, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("ContentIds"), ToolboxData("<{0}:ContentList
runat=server></{0}:ContentList>")>
Public Class ContentList
Inherits EkXsltWebPart
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
Inherits ICallbackEventHandler
```

File

ContentList.cs

ContentList Methods

The methods of the ContentList class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 160)	Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.
	Fill (see page 160)	Populates the data for the ContentList (see page 159).
	GetEnumerator (see page 160)	Gets a System.Collections.IEnumerator for the ContentList (see page 159).
	GetRssFeed (see page 160)	This method returns the ContentList (see page 159) RSS feed.
	MoveNext (see page 160)	Move to next index.

	Reset (see page 161)	Resets the index.
	this (see page 161)	Returns the current ContentItem.ContentRow.
	ToString (see page 161)	Overrides the system ToString function.
	Wrap (see page 161)	Wrap the Html toolbar.

ContentList.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

C#

```
public void CopyTo(Array array, int index);
```

Visual Basic

```
Public Function CopyTo(array As Array, index As Integer) As void
```

ContentList.Fill Method

Populates the data for the ContentList (see page 159).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

ContentList.GetEnumerator Method

Gets a System.Collections.IEnumerator for the ContentList (see page 159).

C#

```
[Browsable(false)]
public IEnumerator GetEnumerator();
```

Visual Basic

```
<Browsable(false)>
Public Function GetEnumerator() As IEnumerator
```

ContentList.GetRssFeed Method

This method returns the ContentList (see page 159) RSS feed.

C#

```
public string GetRssFeed();
```

Visual Basic

```
Public Function GetRssFeed() As string
```

Returns

String

ContentList.MoveNext Method

Move to next index.

C#

```
public bool MoveNext();
```

Visual Basic

```
Public Function MoveNext() As bool
```

Returns

Next Index

ContentList.Reset Method

Resets the index.

C#

```
public void Reset();
```

Visual Basic

```
Public Function Reset() As void
```

ContentList.this Indexer

Returns the current ContentItem.ContentRow.

C#

```
[Browsable(false)]
public Ektron.Cms.Controls.ContentItem.ContentRow this[int i];
```

Visual Basic

```
<Browsable(false)>
Public Sub this(i As Integer)
```

Parameters

Parameters	Description
int i	Index value

Returns

Content (see page 822) Row

ContentList.ToString Method

Overrides the system ToString function.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Returns

String

ContentList.Wrap Method

Wrap the Html toolbar.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	

ContentList Properties

The properties of the ContentList class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 163)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	ContentIds (see page 163)	The comma separated list of content IDs you wish to display.
	Count (see page 164)	Indicates the number of elements in your ContentList (see page 159).
	Current (see page 164)	Gets the current index.
	DefaultContentID (see page 164)	Gets or Sets the Default ContentID for this Control.
	Direction (see page 164)	Direction for sorting results, "ascending" or "descending".
	DynamicParameter (see page 165)	Gets or Sets the QueryString parameter to read a Content (see page 822) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)
	EkItems (see page 165)	Returns the array of ContentBase.
	GetHtml (see page 165)	If set to true, retrieves the content (Html body) of every content block in the content list for display.
	IncludeIcons (see page 166)	Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.
	IsSynchronized (see page 166)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 166)	Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self. • **_Self** = opens in same window • **_Top** = opens in parent window • **_Blank** = opens in new window • **_Parent** = opens in the parent frame
	MarkupLanguage (see page 167)	Enter name of a markup file that manages the display of this server control.
	MetaTag (see page 167)	Gets or Sets the meta tag value. Once this value is set, associated content(s) will be listed.
	OrderByDirection (see page 167)	Determines how to order content in the list. The sort field is determined by the OrderKey (see page 168) property. • ascending - items are arranged A, B, C or 1,2,3. • descending - items are arranged. Z,Y,X or 3,2,1. If sorting by date, descending puts the most recent first. When ascending is selected and the Orderkey property is set to OrderOfTheIds, the order of the IDs are preserved. When set to descending, the order is reversed.

	OrderKey (see page 168)	Sort the list by one of these values: • **Title** - the order of the content's title. • **DateModified** - the last date the content was modified. • **DateCreated** - the date the content was created. • **LastEditorFname** - the last editor's first name. • **LastEditorLname** - the last editor's last name. • **OrderOfTheIds** - preserves the content ID order based on the list in the ContentIds (see page 163) property.
	Random (see page 168)	Set to True if you want to randomly display one content block link from the content list. The content changes each time a user views the page. • **True** - randomly display one content block in the specified folder. • **False** - display the content list normally.
	SuppressWrapperTags (see page 168)	Suppresses the output of span/div tags around the control.
	SyncRoot (see page 169)	Gets an object that can be used to synchronize the ContentList (see page 159).
	XmlDoc (see page 169)	Returns the Xml object loaded for ContentList (see page 159).

ContentList.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable(), WebBrowsable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable(), WebBrowsable()>
Public virtual Property CacheInterval() As double
```

ContentList.ContentIds Property

The comma separated list of content IDs you wish to display.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("The comma separated list of content IDs you wish to display.")]
[Personalizable(), WebBrowsable()]
public string ContentIds;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The comma separated list of content IDs you wish to display.")>
<Personalizable(), WebBrowsable()>
Public Property ContentIds() As string
```

ContentList.Count Property

Indicates the number of elements in your ContentList (see page 159).

C#

```
[Browsable(false)]
public int Count;
```

Visual Basic

```
<Browsable(false)>
Public Property Count() As Integer
```

ContentList.Current Property

Gets the current index.

C#

```
[Browsable(false)]
public object Current;
```

Visual Basic

```
<Browsable(false)>
Public Property Current() As object
```

ContentList.DefaultContentID Property

Gets or Sets the Default ContentID for this Control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Gets or Sets the Default ContentID for this Control.")]
[Personalizable(), WebBrowsable()]
public long DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Gets or Sets the Default ContentID for this Control.")>
<Personalizable(), WebBrowsable()>
Public Property DefaultContentID() As long
```

ContentList.Direction Property

Direction for sorting results, "ascending" or "descending".

C#

```
[Obsolete("Use OrderByDirection Property")]
[Browsable(false)]
[Bindable(true)]
[Category("Ektron")]
[Description("The direction (ascending or descending) that you wish your sorted results to be
```

```
displayed.")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Direction"),
WebDescription("Direction")]
public new Ektron.Cms.Controls.CmsWebService.OrderByDirection Direction;
```

Visual Basic

```
<Obsolete("Use OrderByDirection Property")>
<Browsable(false)>
<Bindable(true)>
<Category("Ektron")>
<Description("The direction (ascending or descending) that you wish your sorted results to be
displayed.")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Direction"),
WebDescription("Direction")>
Public new Property Direction() As Ektron.Cms.Controls.CmsWebService.OrderByDirection
```

ContentList.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 822) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Content ID dynamically. Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Content ID dynamically. Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

ContentList.EkItems Property

Returns the array of ContentBase.

C#

```
[Browsable(false)]
public Ektron.Cms.Common.ContentBase EkItems;
```

Visual Basic

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

ContentList.GetHtml Property

If set to true, retrieves the content (Html body) of every content block in the content list for display.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, retrieves the content (Html body) of every content block in the
content list for display.")]
public bool GetHtml;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true, retrieves the content (Html body) of every content block in the
content list for display.")>
Public Property GetHtml() As bool
```

ContentList.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.

C#

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: Only works with ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
public bool IncludeIcons;
```

Visual Basic

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: Only works with ecmSummary AND ecmTeaser.")>
<DefaultValue(false)>
<Category("Ektron")>
Public Property IncludeIcons() As bool
```

ContentList.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

```
[Browsable(false)]
public bool IsSynchronized;
```

Visual Basic

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

ContentList.LinkTarget Property

Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self.

- **_Self** = opens in same window
- **_Top** = opens in parent window
- **_Blank** = opens in new window
- **_Parent** = opens in the parent frame

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
```

```
Public Property LinkTarget() As ItemLinkTargets
```

ContentList.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

ContentList.MetaTag Property

Gets or Sets the meta tag value. Once this value is set, associated content(s) will be listed.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Gets or Sets the meta tag value.  Once this value is set, associated content(s)
will be listed.")]
public string MetaTag;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Gets or Sets the meta tag value.  Once this value is set, associated content(s)
will be listed.")>
Public Property MetaTag() As string
```

ContentList.OrderByDirection Property

Determines how to order content in the list. The sort field is determined by the OrderKey (see page 168) property.

- ascending - items are arranged A, B, C or 1,2,3.
- descending - items are arranged. Z,Y,X or 3,2,1.

If sorting by date, descending puts the most recent first.

When ascending is selected and the Orderkey property is set to OrderOfTheIds, the order of the IDs are preserved. When set to descending, the order is reversed.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(EkEnumeration.OrderByDirection.Ascending)]
[Description("Direction for sorting results, 'ascending' or 'descending'.")]
public Ektron.Cms.Common.EkEnumeration.OrderByDirection OrderByDirection;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(EkEnumeration.OrderByDirection.Ascending)>
<Description("Direction for sorting results, 'ascending' or 'descending'.")>
Public Property OrderByDirection() As Ektron.Cms.Common.EkEnumeration.OrderByDirection
```

ContentList.OrderKey Property

Sort the list by one of these values:

- **Title** - the order of the content's title.
- **DateModified** - the last date the content was modified.
- **DateCreated** - the date the content was created.
- **LastEditorFname** - the last editor's first name.
- **LastEditorLname** - the last editor's last name.
- **OrderOfTheIds** - preserves the content ID order based on the list in the ContentIds (see page 163) property.

C#

```
[Bindable(true)]
[DefaultValue(Ektron.Cms.Controls.CmsWebService.ContentListOrderBy.OrderOfTheIds)]
[Description("Sort the list by one of these values: Title, DateModified, DateCreated,
LastEditorFname, LastEditorLname or OrderOfTheIds.")]
[Category("Ektron")]
public Ektron.Cms.Controls.CmsWebService.ContentListOrderBy OrderKey;
```

Visual Basic

```
<Bindable(true)>
<DefaultValue(Ektron.Cms.Controls.CmsWebService.ContentListOrderBy.OrderOfTheIds)>
<Description("Sort the list by one of these values: Title, DateModified, DateCreated,
LastEditorFname, LastEditorLname or OrderOfTheIds.")>
<Category("Ektron")>
Public Property OrderKey() As Ektron.Cms.Controls.CmsWebService.ContentListOrderBy
```

ContentList.Random Property

Set to True if you want to randomly display one content block link from the content list. The content changes each time a user views the page.

- **True** - randomly display one content block in the specified folder.
- **False** - display the content list normally.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Set this property to True to grab a single, random piece of content from the
content list.")]
public bool Random;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Set this property to True to grab a single, random piece of content from the
content list.")>
Public Property Random() As bool
```

ContentList.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

C#

```
public override bool SuppressWrapperTags;
```

Visual Basic

```
Public override Property SuppressWrapperTags() As bool
```

ContentList.SyncRoot Property

Gets an object that can be used to synchronize the ContentList (see page 159).

C#

```
[Browsable(false)]
public object SyncRoot;
```

Visual Basic

```
<Browsable(false)>
Public Property SyncRoot() As object
```

ContentList.XmlDoc Property

Returns the Xml object loaded for ContentList (see page 159).

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

ContentRating Class

Use the ContentRating server control to place a rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. A visitor can rank the content on a numerical scale of one through nine, as well as submit a text comment.

Class Hierarchy

C#

```
[DefaultProperty("DefaultContentID"), ToolboxData("<{0}:ContentRating
runat=server></{0}:ContentRating>")]
public class ContentRating : EkWebControl;
```

Visual Basic

```
<DefaultProperty("DefaultContentID"), ToolboxData("<{0}:ContentRating
runat=server></{0}:ContentRating>")>
Public Class ContentRating
Inherits EkWebControl
```

File

ContentRating.cs

Description

[Obsolete()]

ContentRating Methods

The methods of the ContentRating class are listed here.

Public Methods

	Name	Description
	Fill (see page 170)	Method that can be overridden and used to grab data from the back end for each server control.

	ToString (see page 170)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 170)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

ContentRating.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

ContentRating.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

ContentRating.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

ContentRating Properties

The properties of the ContentRating class are listed here.

Public Properties

	Name	Description
	AlreadyRatedMessage (see page 171)	The label displayed to a user thanking them for their content rating. Defaults to "Thank you for your feedback".
	BadLabel (see page 171)	The label corresponding to a "Bad" content rating. This defaults to "Poor".
	CharsRemainingLabel (see page 172)	Sets or gets the label for the number of additional characters a user may enter into the comment. The default is "Characters Remaining:"
	CompleteGraphDisplay (see page 172)	Described the layout manner which the graph is displayed to a user after content rating. The graph will be displayed to the right of the dialog in the event of Horizontal, below in the case of Vertical, and not displayed in the case of None.
	CookiesRequiredMessage (see page 172)	The message that appears when a user needs to enable cookies on their system. The default is: "We're sorry, but you must enable cookies in your browser to rate this content."
	DefaultContentID (see page 173)	Gets or Sets the Default ContentID for this Control. This is the ContentID that will be displayed if the DynamicParameter (see page 173) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 173) of "id", and will display DefaultContentID otherwise.)

	DynamicParameter (see page 173)	Gets or Sets the QueryString parameter to read a Content (see page 822) ID dynamically. Set to "" (empty) if you wish to always display the default ContentID (static.)
	FeedbackHeader (see page 173)	The label corresponding to how the user would rate the content block. Defaults to "How would you rate the usefulness of this content?".
	GoodLabel (see page 174)	The label corresponding to a "Good" content rating. This defaults to "Outstanding".
	GraphBarColor (see page 174)	The color of the bars in the Content (see page 822) Rating bar graph.
	InitialGraphDisplay (see page 174)	Describes the layout manner which the graph is displayed to a user before content rating. The graph will be displayed to the right of the dialog in the event of Horizontal, below in the case of Vertical, and not displayed in the case of None.
	Padding (see page 175)	Padding, in pixels, to be applied between the content rating input and display graph.
	RatingLevelLabel (see page 175)	The label corresponding to a "Good" content rating. This defaults to "Outstanding".
	ThankYouMessage (see page 176)	The label displayed to a user thanking them for their content rating. Defaults to "Thank you for your feedback.".
	TotalRatingsLabel (see page 176)	The label that appears below the graph, and lists the number of rating submissions received. The default is: "Total Ratings".
	UserCommentsHeader (see page 176)	The label corresponding to why the user rated content the way they did. Defaults to "Tell us why you rated the content this way".
	ValidationMessage (see page 177)	The label displayed to a user when they try to submit a content rating without selecting a value of 1-9. Defaults to "Please select a rating before submitting your feedback.".

ContentRating.AlreadyRatedMessage Property

The label displayed to a user thanking them for their content rating. Defaults to "Thank you for your feedback".

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label displayed to a user thanking them for their content rating. Defaults
to \"Thank you for your feedback.\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_RATED)]
public string AlreadyRatedMessage;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label displayed to a user thanking them for their content rating. Defaults
to \"Thank you for your feedback.\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_RATED)>
Public Property AlreadyRatedMessage() As string
```

ContentRating.BadLabel Property

The label corresponding to a "Bad" content rating. This defaults to "Poor".

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label corresponding to a \"Bad\" content rating. This defaults to
\"Poor\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_BAD)]
public string BadLabel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label corresponding to a \"Bad\" content rating. This defaults to
\"Poor\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
```

```
<DefaultValue(EkWebControl.RES_CNTRATE_BAD)>
Public Property BadLabel() As string
```

ContentRating.CharsRemainingLabel Property

Sets or gets the label for the number of additional characters a user may enter into the comment. The default is "Characters Remaining:"

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("Sets or gets the label for the number of additional characters a user may enter
into the comment. The default is Characters Remaining:")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_CHARS)]
public string CharsRemainingLabel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("Sets or gets the label for the number of additional characters a user may enter
into the comment. The default is Characters Remaining:")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_CHARS)>
Public Property CharsRemainingLabel() As string
```

ContentRating.CompleteGraphDisplay Property

Described the layout manner which the graph is displayed to a user after content rating. The graph will be displayed to the right of the dialog in the event of Horizontal, below in the case of Vertical, and not displayed in the case of None.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Described the layout manner which the graph is displayed to a user after content
rating. The graph will be displayed to the right of the dialog in the event of Horizontal,
below in the case of Verical, and not displayed in the case of None.")]
[DefaultValue(Ektron.Cms.ContentAPI.RatingGraphDisplayType.Horizontal)]
public Ektron.Cms.ContentAPI.RatingGraphDisplayType CompleteGraphDisplay;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Described the layout manner which the graph is displayed to a user after content
rating. The graph will be displayed to the right of the dialog in the event of Horizontal,
below in the case of Verical, and not displayed in the case of None.")>
<DefaultValue(Ektron.Cms.ContentAPI.RatingGraphDisplayType.Horizontal)>
Public Property CompleteGraphDisplay() As Ektron.Cms.ContentAPI.RatingGraphDisplayType
```

ContentRating.CookiesRequiredMessage Property

The message that appears when a user needs to enable cookies on their system. The default is: "We're sorry, but you must enable cookies in your browser to rate this content."

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The message that appears when a user needs to enable cookies on their system.
The default is: \"We're sorry, but you must enable cookies in your browser to rate this
content.\"")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_COOK)]
public string CookiesRequiredMessage;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The message that appears when a user needs to enable cookies on their system.
The default is: \"We're sorry, but you must enable cookies in your browser to rate this
content.\"")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_COOK)>
Public Property CookiesRequiredMessage() As string
```

ContentRating.DefaultContentID Property

Gets or Sets the Default ContentID for this Control. This is the ContentID that will be displayed if the DynamicParameter (see page 173) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 173) of "id", and will display DefaultContentID otherwise.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentRatingUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the Default ContentID for this Control. This is the ContentID that
will be displayed if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will get
the value 2 for a DynamicParameter of \"id\", and will display DefaultContentID otherwise.)")]
public long DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentRatingUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the Default ContentID for this Control. This is the ContentID that
will be displayed if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will get
the value 2 for a DynamicParameter of \"id\", and will display DefaultContentID otherwise.)")>
Public Property DefaultContentID() As long
```

ContentRating.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 822) ID dynamically. Set to "" (empty) if you wish to always display the default ContentID (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Content ID dynamically. Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Content ID dynamically. Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

ContentRating.FeedbackHeader Property

The label corresponding to how the user would rate the content block. Defaults to "How would you rate the usefulness of this

content?".

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label corresponding to how the user would rate the content block. Defaults
to \"How would you rate the usefullness of this content?\"")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_FEED)]
public string FeedbackHeader;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label corresponding to how the user would rate the content block. Defaults
to \"How would you rate the usefullness of this content?\"")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_FEED)>
Public Property FeedbackHeader() As string
```

ContentRating.GoodLabel Property

The label corresponding to a "Good" content rating. This defaults to "Outstanding".

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label corresponding to a \"Good\" content rating. This defaults to
\"Ouotstanding\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_GOOD)]
public string GoodLabel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label corresponding to a \"Good\" content rating. This defaults to
\"Ouotstanding\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_GOOD)>
Public Property GoodLabel() As string
```

ContentRating.GraphBarColor Property

The color of the bars in the Content (see page 822) Rating bar graph.

C#

```
[Category("Ektron")]
[Bindable(true)]
[DefaultValue(typeof(System.Drawing.Color), "LightBlue")]
[Description("The color of the bars in the Content Rating bar graph.")]
public System.Drawing.Color GraphBarColor;
```

Visual Basic

```
<Category("Ektron")>
<Bindable(true)>
<DefaultValue(typeof(System.Drawing.Color), "LightBlue")>
<Description("The color of the bars in the Content Rating bar graph.")>
Public Property GraphBarColor() As System.Drawing.Color
```

ContentRating.InitialGraphDisplay Property

Describes the layout manner which the graph is displayed to a user before content rating. The graph will be displayed to the right

of the dialog in the event of Horizontal, below in the case of Vertical, and not displayed in the case of None.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Described the layout manner which the graph is displayed to a user before
content rating. The graph will be displayed to the right of the dialog in the event of
Horizontal, below in the case of Verical, and not displayed in the case of None.")]
[DefaultValue(Ektron.Cms.ContentAPI.RatingGraphDisplayType.Horizontal)]
public Ektron.Cms.ContentAPI.RatingGraphDisplayType InitialGraphDisplay;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Described the layout manner which the graph is displayed to a user before
content rating. The graph will be displayed to the right of the dialog in the event of
Horizontal, below in the case of Verical, and not displayed in the case of None.")>
<DefaultValue(Ektron.Cms.ContentAPI.RatingGraphDisplayType.Horizontal)>
Public Property InitialGraphDisplay() As Ektron.Cms.ContentAPI.RatingGraphDisplayType
```

ContentRating.Padding Property

Padding, in pixels, to be applied between the content rating input and display graph.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("Padding, in pixels, to be applied between the content rating input and display
graph.")]
public int Padding;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("Padding, in pixels, to be applied between the content rating input and display
graph.")>
Public Property Padding() As Integer
```

ContentRating.RatingLevelLabel Property

The label corresponding to a "Good" content rating. This defaults to "Outstanding".

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label corresponding to a \"Good\" content rating. This defaults to
\"Ouotstanding\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_LEVEL)]
public string RatingLevelLabel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label corresponding to a \"Good\" content rating. This defaults to
\"Ouotstanding\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_LEVEL)>
Public Property RatingLevelLabel() As string
```

ContentRating.ThankYouMessage Property

The label displayed to a user thanking them for their content rating. Defaults to "Thank you for your feedback.".

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label displayed to a user thanking them for their content rating. Defaults
to \"Thank you for your feedback.\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_THANK)]
public string ThankYouMessage;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label displayed to a user thanking them for their content rating. Defaults
to \"Thank you for your feedback.\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_THANK)>
Public Property ThankYouMessage() As string
```

ContentRating.TotalRatingsLabel Property

The label that appears below the graph, and lists the number of rating submissions received. The default is: "Total Ratings".

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label that appears below the graph, and lists the number of rating
submissions received. The default is: \"Total Ratings\"")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_TOTAL)]
public string TotalRatingsLabel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label that appears below the graph, and lists the number of rating
submissions received. The default is: \"Total Ratings\"")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_TOTAL)>
Public Property TotalRatingsLabel() As string
```

ContentRating.UserCommentsHeader Property

The label corresponding to why the user rated content the way they did. Defaults to "Tell us why you rated the content this way".

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label corresponding to why the user rated content the way they did. Defaults
to \"Tell us why you rated the content this way\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_CMNT)]
public string UserCommentsHeader;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label corresponding to why the user rated content the way they did. Defaults
to \"Tell us why you rated the content this way\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
```

```
<DefaultValue(EkWebControl.RES_CNTRATE_CMNT)>
Public Property UserCommentsHeader() As string
```

ContentRating.ValidationMessage Property

The label displayed to a user when they try to submit a content rating without selecting a value of 1-9. Defaults to "Please select a rating before submitting your feedback.".

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label displayed to the user when they try to submit a content rating without
selecting a value of 1-9. Defaults to \"Please select a rating before submitting your
feedback.\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_VALID)]
public string ValidationMessage;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label displayed to the user when they try to submit a content rating without
selecting a value of 1-9. Defaults to \"Please select a rating before submitting your
feedback.\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_VALID)>
Public Property ValidationMessage() As string
```

ContentReview Class

The ContentReview server control has two ways it can be used. The first, places a star based rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. An item is rated when a visitor clicks on a star. Depending on the XSLT chosen, visitors can also submit review comments.

The second way the ContentReview control can be used is to retrieve and display visitor reviews and comments. In this mode, the control can display reviews of a content item or reviews by a specific site or membership user.

Class Hierarchy

C#

```
[DefaultProperty("DefaultContentID"), ToolboxData("<{0}:ContentReview
runat=server></{0}:ContentReview>")]
public class ContentReview : EkXsltWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultContentID"), ToolboxData("<{0}:ContentReview
runat=server></{0}:ContentReview>")>
Public Class ContentReview
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

File

ContentReview.cs

ContentReview Methods

The methods of the ContentReview class are listed here.

Public Methods

	Name	Description
	Fill (see page 178)	Method that can be overridden and used to grab data from the back end for each server control.
	RaiseCallbackEvent (see page 178)	
	ToString (see page 178)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 178)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

ContentReview.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

ContentReview.RaiseCallbackEvent Method

C#

```
public void RaiseCallbackEvent(string eventArgs);
```

Visual Basic

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

Parameters

Parameters	Description
string eventArgs	string

ContentReview.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

ContentReview.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

ContentReview Properties

The properties of the ContentReview class are listed here.

Public Properties

	Name	Description
	DefaultContentID (see page 179)	Gets or Sets the default content ID for this control. This is the content ID that will be rated if the DynamicParameter (see page 180) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 180) of "id", and will display DefaultContentID otherwise.)
	DisplayXslt (see page 180)	Select the type of review to display, or enter the path to a custom XSLT. (optional). Choices are: • **Ajax 5 Stars** - a five star rating system utilizing Ajax for display. • **Ajax 5 Stars Comment** - a five star rating system utilizing Ajax for display. In addition, when you hover over the stars, a review box pops-up below the stars. Comments that are added are submitted via an Ajax call. • **Ajax 5 Stars with Increments** - a five star rating system with half star increments that utilizes Ajax for display. • **5 Stars** - a five star rating system that allows... more (see page 180)
	DynamicParameter (see page 180)	To make this content review control dynamic, select id. When you do, this server control is attached to the content block passed as a URL parameter. When using this control to retrieve a user's reviews, set this property to UserId to make the User (see page 950)'s ID dynamic.
	DynamicUserParameter (see page 181)	The querystring parameter for which to get the user ID. If a value is found in this parameter, it is used, or the UserID (see page 182) property is used. Important: The GetReviews (see page 181) property must be set to User (see page 950) for the control to use this property.
	GetReviews (see page 181)	This property returns a list of reviews for a content item or a specified user. It is used to decide which type of review information is retrieved by the server control. To display reviews on a page, you must use an XSLT that is set up for displaying reviews.
	MaxReviews (see page 181)	The number of reviews to retrieve if the GetReview property is set to Content (see page 822) or User (see page 950). 0 (zero) = unlimited results.
	Moderate (see page 182)	Setting this property to True allows CMS400.NET users to moderate reviews. Additional information on moderating reviews can be found in the Ektron CMS400.NET Administrator Manual section "Managing HTML Content (see page 822)" > "Content (see page 822) Rating"
	RatingsMinimum (see page 182)	Sets a minimum number of the reviews and ratings before displaying the average rating.
	UserID (see page 182)	The user ID for which to get reviews. If left blank, reviews from all users are returned. If a user ID is specified, only reviews for that user are returned. Important: The GetReviews (see page 181) property must be set to User (see page 950) for the control to use this property.
	XmlDoc (see page 183)	Returns the Xslt object for the collection.

ContentReview.DefaultContentID Property

Gets or Sets the default content ID for this control. This is the content ID that will be rated if the DynamicParameter (see page 180) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 180) of "id", and will display DefaultContentID otherwise.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentReviewUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the default content ID for this control. This is the content ID
that will be rated if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will
get the value 2 for a DynamicParameter of \"id\", and will review DefaultContentID
otherwise.)")]
public long DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
```

```
<Category("Ektron")>
<Editor(typeof(ContentReviewUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the default content ID for this control. This is the content ID
that will be rated if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will
get the value 2 for a DynamicParameter of \"id\", and will review DefaultContentID
otherwise.)")>
Public Property DefaultContentID() As long
```

ContentReview.DisplayXslt Property

Select the type of review to display, or enter the path to a custom XSLT. (optional).

Choices are:

- **Ajax 5 Stars** - a five star rating system utilizing Ajax for display.
- **Ajax 5 Stars Comment** - a five star rating system utilizing Ajax for display. In addition, when you hover over the stars, a review box pops-up below the stars. Comments that are added are submitted via an Ajax call.
- **Ajax 5 Stars with Increments** - a five star rating system with half star increments that utilizes Ajax for display.
- **5 Stars** - a five star rating system that allows visitors to add text reviews of content.
- **5 Stars with Increments** - a five star rating system with half star increments that allows visitors to add text reviews of content.
- **Review List** - displays a list of reviews for content or a user. See Also:
- **Path to Custom Xslt** - enter the path to a custom XSLT.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the Control (optional)")]
[TypeConverter(typeof(XsltContentReviewTypeConverter))]
[DefaultValue("Ajax 5 Stars with increments")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the Control (optional)")>
<TypeConverter(typeof(XsltContentReviewTypeConverter))>
<DefaultValue("Ajax 5 Stars with increments")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

ContentReview.DynamicParameter Property

To make this content review control dynamic, select id. When you do, this server control is attached to the content block passed as a URL parameter. When using this control to retrieve a user's reviews, set this property to UserId to make the User (see page 950)'s ID dynamic.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("To make this content review control dynamic, select id. When you do, this server
control is attached to the content block passed as a URL parameter. When using this control to
retrieve a user?s reviews, set this property to UserId to make the user?s ID dynamic.")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
```

```
<Category("Ektron")>
<Description("To make this content review control dynamic, select id. When you do, this server
control is attached to the content block passed as a URL parameter. When using this control to
retrieve a user?s reviews, set this property to UserId to make the user?s ID dynamic.")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

ContentReview.DynamicUserParameter Property

The querystring parameter for which to get the user ID. If a value is found in this parameter, it is used, or the UserID (see page 182) property is used. Important: The GetReviews (see page 181) property must be set to User (see page 950) for the control to use this property.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The querystring parameter for which to get the user ID. If a value is found in
this parameter, it is used, or the UserID property is used. Important: The GetReviews property
must be set to User for the control to use this property.")]
[TypeConverter(typeof(DynamicUserIDTypeConverter))]
[DefaultValue("userid")]
public string DynamicUserParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The querystring parameter for which to get the user ID. If a value is found in
this parameter, it is used, or the UserID property is used. Important: The GetReviews property
must be set to User for the control to use this property.")>
<TypeConverter(typeof(DynamicUserIDTypeConverter))>
<DefaultValue("userid")>
Public Property DynamicUserParameter() As string
```

ContentReview.GetReviews Property

This property returns a list of reviews for a content item or a specified user. It is used to decide which type of review information is retrieved by the server control. To display reviews on a page, you must use an XSLT that is set up for displaying reviews.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("This property returns a list of reviews for a content item or a specified user.
It is used to decide which type of review information is retrieved by the server control. To
display reviews on a page, you must use an XSLT that is set up for displaying reviews.")]
[DefaultValue(ReviewTypes.none)]
public ReviewTypes GetReviews;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("This property returns a list of reviews for a content item or a specified user.
It is used to decide which type of review information is retrieved by the server control. To
display reviews on a page, you must use an XSLT that is set up for displaying reviews.")>
<DefaultValue(ReviewTypes.none)>
Public Property GetReviews() As ReviewTypes
```

ContentReview.MaxReviews Property

The number of reviews to retrieve if the GetReview property is set to Content (see page 822) or User (see page 950). 0 (zero) = unlimited results.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The number of reviews to retrieve if the GetReview property is set to Content or
User. 0  = unlimited results.")]
[DefaultValue(0)]
public int MaxReviews;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The number of reviews to retrieve if the GetReview property is set to Content or
User. 0  = unlimited results.")>
<DefaultValue(0)>
Public Property MaxReviews() As Integer
```

ContentReview.Moderate Property

Setting this property to True allows CMS400.NET users to moderate reviews.

Additional information on moderating reviews can be found in the Ektron CMS400.NET Administrator Manual section “Managing HTML Content (see page 822)” > “Content (see page 822) Rating”

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Setting this property to True allows CMS400.NET users to moderate reviews.")]
[DefaultValue(false)]
public bool Moderate;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Setting this property to True allows CMS400.NET users to moderate reviews.")>
<DefaultValue(false)>
Public Property Moderate() As bool
```

ContentReview.RatingsMinimum Property

Sets a minimum number of the reviews and ratings before displaying the average rating.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets a minimum number of the reviews and ratings before displaying the average
rating.")]
[DefaultValue(0)]
public int RatingsMinimum;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets a minimum number of the reviews and ratings before displaying the average
rating.")>
<DefaultValue(0)>
Public Property RatingsMinimum() As Integer
```

ContentReview.UserID Property

The user ID for which to get reviews. If left blank, reviews from all users are returned. If a user ID is specified, only reviews for that user are returned.

Important: The GetReviews (see page 181) property must be set to User (see page 950) for the control to use this property.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The user ID for which to get reviews. If left blank, reviews from all users are
returned. If a user ID is specified, only reviews for that user are returned. Important: The
GetReviews property must be set to User for the control to use this property.")]
[DefaultValue(0)]
public long UserID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The user ID for which to get reviews. If left blank, reviews from all users are
returned. If a user ID is specified, only reviews for that user are returned. Important: The
GetReviews property must be set to User for the control to use this property.")>
<DefaultValue(0)>
Public Property UserID() As long
```

ContentReview.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

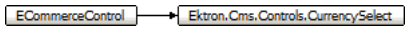
Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

CurrencySelect Class

Summary description for CurrencySelect control

Class Hierarchy



C#

```
[ToolboxData("<{0}:CurrencySelect runat=server></{0}:CurrencySelect>")]
public class CurrencySelect : ECommerceControl;
```

Visual Basic

```
<ToolboxData("<{0}:CurrencySelect runat=server></{0}:CurrencySelect>")>
Public Class CurrencySelect
Inherits ECommerceControl
```

File

CurrencySelect.cs

CurrencySelect Methods

The methods of the CurrencySelect class are listed here.

Public Methods

	Name	Description
	Fill (see page 184)	Populates the data.
	ToString (see page 184)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 184)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

CurrencySelect.Fill Method

Populates the data.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

CurrencySelect.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

CurrencySelect.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

CurrencySelect Properties

The properties of the CurrencySelect class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 184)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DisplayXslt (see page 185)	Specify an external XSLT file to render the control. By default, the control uses CurrencySelect.xsl. This file is located in:
	EkItem (see page 186)	
	Stylesheet (see page 186)	Specify the path to a style sheet for use with this server control. The location can be relative or absolute. Leave blank to use the default style sheet.

CurrencySelect.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public new virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public new virtual Property CacheInterval() As double
```

CurrencySelect.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses CurrencySelect.xsl. This file is located in:

```
[Site Root]\Workarea\Xslt\Commerce
```

Important: If you want to edit this file, it is strongly recommended that you make a copy, change its name and move it outside of the Workarea folder. If you make changes to this file and do not move it out of the Workarea, this file and the changes will be lost when you upgrade.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses CurrencySelect.xsl. This file is located in \"<Site Root>\\Workarea\\Xslt\\Commerce.\"
Important: If you want to edit this file, it is strongly recommended that you make a copy,
change its name and move it outside of the Workarea folder. If you make changes to this file
and do not move it out of the Workarea, this file and the changes will be lost when you
upgrade.")]
[TypeConverter(typeof(XsltOrderTypeConverter))]
[DefaultValue("Commerce/CurrencySelect.xsl")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses CurrencySelect.xsl. This file is located in \"<Site Root>\\Workarea\\Xslt\\Commerce.\"
Important: If you want to edit this file, it is strongly recommended that you make a copy,
change its name and move it outside of the Workarea folder. If you make changes to this file
and do not move it out of the Workarea, this file and the changes will be lost when you
upgrade.")>
<TypeConverter(typeof(XsltOrderTypeConverter))>
<DefaultValue("Commerce/CurrencySelect.xsl")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

CurrencySelect.EkItem Property

C#

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Returns Currency Data.")]
[Personalizable()]
public List<CurrencyData> EkItem;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Returns Currency Data.")>
<Personalizable()>
Public Property EkItem() As List<CurrencyData>
```

CurrencySelect.Stylesheet Property

Specify the path to a style sheet for use with this server control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with this server control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[TypeConverter(typeof(CurrencySelectCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/CurrencySelect.css")]
public string Stylesheet;
```

Visual Basic

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with this server control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<TypeConverter(typeof(CurrencySelectCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/CurrencySelect.css")>
Public Property Stylesheet() As string
```

DesignTimeDiagnostic Class

The DesignTimeDiagnostic server control is used to verify the connection to Ektron's server controls Web service. When added to a Web form, this control provides the following information:

- **Web Service Path** - displays the server control Web service path in your web.config file
- **Web Service Exists** - calls a method in the Web service that returns True when the Web service exists. When it does not, it returns False.
- **Test Login (see page 347) Control** - displays the Login (see page 347) server control to show that the Web service is connected and working properly. When not connecting properly, an error message is displayed.

Class Hierarchy

C#

```
[ToolboxData("<{0}:DesignTimeDiagnostic runat=server></{0}:DesignTimeDiagnostic>")]
public class DesignTimeDiagnostic : EkWebControl;
```

Visual Basic

```
<ToolboxData("<{0}:DesignTimeDiagnostic runat=server></{0}:DesignTimeDiagnostic>")>
```

```
Public Class DesignTimeDiagnostic
Inherits EkWebControl
```

File

DesignTimeDiagnostic.cs

DesignTimeDiagnostic Methods

The methods of the DesignTimeDiagnostic class are listed here.

Public Methods

	Name	Description
	Fill (see page 187)	Method that can be overridden and used to grab data from the back end for each server control.

DesignTimeDiagnostic.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

DesignTimeDiagnostic Properties

The properties of the DesignTimeDiagnostic class are listed here.

Public Properties

	Name	Description
	Authenticated (see page 187)	DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and permits them to login through a login dialog in Visual Studio.NET.
	Hide (see page 187)	Alias for the 'Visible' property. Visible is set to mirror this value. This property is used to hide output in design time, whereas Visible does not hide output in design time.
	Language (see page 188)	Set a language for viewing content blocks, collections, etc. in different languages. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).
	SuppressWrapperTags (see page 188)	Suppresses the output of span/div tags around the control.
	WrapTag (see page 188)	Allows a developer to specify control's tag. Default is assumed "".

DesignTimeDiagnostic.Authenticated Property

DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and permits them to login through a login dialog in Visual Studio.NET.

C#

```
[Browsable(false)]
public new string Authenticated;
```

Visual Basic

```
<Browsable(false)>
Public new Property Authenticated() As string
```

DesignTimeDiagnostic.Hide Property

Alias for the 'Visible' property. Visible is set to mirror this value. This property is used to hide output in design time, whereas Visible does not hide output in design time.

C#

```
[Browsable(false)]
public new bool Hide;
```

Visual Basic

```
<Browsable(false)>
Public new Property Hide() As bool
```

DesignTimeDiagnostic.Language Property

Set a language for viewing content blocks, collections, etc. in different languages. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).

C#

```
[Browsable(false)]
public new int Language;
```

Visual Basic

```
<Browsable(false)>
Public new Property Language() As Integer
```

DesignTimeDiagnostic.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

C#

```
[Browsable(false)]
public new bool SuppressWrapperTags;
```

Visual Basic

```
<Browsable(false)>
Public new Property SuppressWrapperTags() As bool
```

DesignTimeDiagnostic.WrapTag Property

Allows a developer to specify control's tag. Default is assumed "".

C#

```
[Browsable(false)]
public new string WrapTag;
```

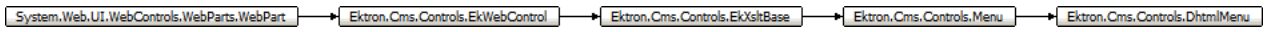
Visual Basic

```
<Browsable(false)>
Public new Property WrapTag() As string
```

DhtmlMenu Class

The DhtmlMenu server control lets you display a menu with drop down boxes and submenus. This lets developers add a menu to a page without taking up a lot space.

Class Hierarchy



C#

```
[DefaultProperty("DefaultMenuID")]
public class DhtmlMenu : Ektron.Cms.Controls.Menu;
```

Visual Basic

```
<DefaultProperty("DefaultMenuID")>
```

```
Public Class DhtmlMenu
Inherits Ektron.Cms.Controls.Menu
```

File

DhtmlMenu.cs

DhtmlMenu Methods

The methods of the DhtmlMenu class are listed here.

Public Methods

	Name	Description
	Fill (see page 189)	Calls up the database and updates the contents of the Menu (see page 368) if the data is out of sync with the specified parameters.
	Wrap (see page 189)	Method that can be overridden and used to wrap an Ektron Toolbar corresponding to a particular Control around some Html text.

DhtmlMenu.Fill Method

Calls up the database and updates the contents of the Menu (see page 368) if the data is out of sync with the specified parameters.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

DhtmlMenu.Wrap Method

Method that can be overridden and used to wrap an Ektron Toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

Html string wrapped with a toolbar.

DhtmlMenu Properties

The properties of the DhtmlMenu class are listed here.

Public Properties

	Name	Description
	AccessKey (see page 190)	Ignored. The property has no effect.

	Align (see page 190)	Specify the horizontal alignment for the root menu text. The default is Center. Choices are: • **NotSet** - Use the default setting. • **Left** - Aligns text to the left. • **Center** - Aligns text to the center. • **Right** - Aligns text to the right. • **Justify** - Aligns text justified to both margins.
	BackColor (see page 191)	BackColor sets the menu background color.
	BorderColor (see page 191)	Sets the border color around the editor.
	BorderStyle (see page 191)	Sets the border style around the editor.
	BorderWidth (see page 191)	Sets the border width around the editor.
	CssClass (see page 192)	Apply a .css class to the root menu. Enter the class name from the .css defined in the Stylesheet (see page 194) property. Since pop_style.css is the default style sheet, you can use its classes without defining it in the Stylesheet (see page 194) property.
	DisplayXslt (see page 192)	Ignored. The property has no effect.
	Enabled (see page 192)	Ignored. The property has no effect.
	EnableIE6FormsFix (see page 192)	This should only be enabled when you have a DHTML menu rendering over an HTML Select element. In IE6, the DHTML will render below the drop down. This will temporarily disable the element when the DHTML menu is rendered.
	Height (see page 193)	The height (in pixels or percent) of the editor, including toolbars.
	HorizontalDisplay (see page 193)	Popup root menu horizontally.
	MenuHoverOverColor (see page 193)	`HoverOverColor` sets the menu hoverover color.
	MenuStyle (see page 194)	Specify the custom style to use. If specified, all specified appearance properties will be ignored.
	Stylesheet (see page 194)	Specify the style sheet relative to site root. Leave blank for default.
	SubMenuBackColor (see page 194)	`SubMenuBackColor` sets the sub menu background color.
	SubMenuHoverOverColor (see page 194)	`SubMenuHoverOverColor` sets the menu hoverover color.
	SubMenuItemHeight (see page 195)	`SubMenuItemHeight` is the height in pixels of submenu items.
	SubMenuItemWidth (see page 195)	`SubMenuItemWidth` is the width in pixels of submenu items.
	SubMenuTopOffset (see page 195)	`SubMenuTopOffset` sets the horizontal offset of submenu popups relative to the parent submenu item.
	TabIndex (see page 196)	Ignored. The property has no effect.
	ToolTip (see page 196)	Ignored. The property has no effect.
	Width (see page 196)	The width (in pixels or percent) of the editor, including toolbars.

DhtmlMenu.AccessKey Property

Ignored. The property has no effect.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override string AccessKey;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property AccessKey() As string
```

DhtmlMenu.Align Property

Specify the horizontal alignment for the root menu text. The default is Center. Choices are:

- **NotSet** - Use the default setting.
- **Left** - Aligns text to the left.

- **Center** - Aligns text to the center.
- **Right** - Aligns text to the right.
- **Justify** - Aligns text justified to both margins.

C#

```
[Description("Specify the horizontal alignment for the root menu text. Choices are: NotSet,
Left, Center, Right or Justify.")]
[Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(HorizontalAlign))]
public HorizontalAlign Align;
```

Visual Basic

```
<Description("Specify the horizontal alignment for the root menu text. Choices are: NotSet,
Left, Center, Right or Justify.")>
<Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(HorizontalAlign))>
Public Property Align() As HorizontalAlign
```

DhtmlMenu.BackColor Property

BackColor sets the menu background color.

C#

```
[NotifyParentProperty(true), Browsable(true), Bindable(true)]
[Description("BackColor sets the menu background color.")]
public override System.Drawing.Color BackColor;
```

Visual Basic

```
<NotifyParentProperty(true), Browsable(true), Bindable(true)>
<Description("BackColor sets the menu background color.")>
Public override Property BackColor() As System.Drawing.Color
```

DhtmlMenu.BorderColor Property

Sets the border color around the editor.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override System.Drawing.Color BorderColor;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property BorderColor() As System.Drawing.Color
```

DhtmlMenu.BorderStyle Property

Sets the border style around the editor.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override BorderStyle BorderStyle;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property BorderStyle() As BorderStyle
```

DhtmlMenu.BorderWidth Property

Sets the border width around the editor.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override Unit BorderWidth;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property BorderWidth() As Unit
```

DhtmlMenu.CssClass Property

Apply a .css class to the root menu. Enter the class name from the .css defined in the Stylesheet (see page 194) property. Since pop_style.css is the default style sheet, you can use its classes without defining it in the Stylesheet (see page 194) property.

C#

```
[Description("Specify the class name to use instead of internal style."), Category("Ektron")]
public override string CssClass;
```

Visual Basic

```
<Description("Specify the class name to use instead of internal style."), Category("Ektron")>
Public override Property CssClass() As string
```

DhtmlMenu.DisplayXslt Property

Ignored. The property has no effect.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override string DisplayXslt;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property DisplayXslt() As string
```

DhtmlMenu.Enabled Property

Ignored. The property has no effect.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override bool Enabled;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property Enabled() As bool
```

DhtmlMenu.EnableIE6FormsFix Property

This should only be enabled when you have a DHTML menu rendering over an HTML Select element. In IE6, the DHTML will render below the drop down. This will temporarily disable the element when the DHTML menu is rendered.

C#

```
[Browsable(true)]
[Description("This should only be enabled when you have a DHTML menu rendering over an HTML
Select element. In IE6, the DHTML will render below the drop down. This will temporarily
disable the element when the DHTML menu is rendered.")]
[Bindable(true)]
[Category("Ektron")]
public bool EnableIE6FormsFix;
```

Visual Basic

```
<Browsable(true)>
<Description("This should only be enabled when you have a DHTML menu rendering over an HTML
Select element. In IE6, the DHTML will render below the drop down. This will temporarily
disable the element when the DHTML menu is rendered.")>
<Bindable(true)>
<Category("Ektron")>
Public Property EnableIE6FormsFix() As bool
```

DhtmlMenu.Height Property

The height (in pixels or percent) of the editor, including toolbars.

C#

```
[NotifyParentProperty(true), Browsable(true), Bindable(true), Category("Layout"),
DefaultValue(m_DefaultHeight)]
[Description("The height (in pixels or percent) of the editor, including toolbars.")]
public override Unit Height;
```

Visual Basic

```
<NotifyParentProperty(true), Browsable(true), Bindable(true), Category("Layout"),
DefaultValue(m_DefaultHeight)>
<Description("The height (in pixels or percent) of the editor, including toolbars.")>
Public override Property Height() As Unit
```

DhtmlMenu.HorizontalDisplay Property

Popup root menu horizontally.

C#

```
[Description("Popup root menu horizontally."), Category("Ektron")]
[Browsable(true)]
[Bindable(false)]
[DefaultValue(false)]
public bool HorizontalDisplay;
```

Visual Basic

```
<Description("Popup root menu horizontally."), Category("Ektron")>
<Browsable(true)>
<Bindable(false)>
<DefaultValue(false)>
Public Property HorizontalDisplay() As bool
```

DhtmlMenu.MenuHoverOverColor Property

C#

```
[Description("HoverOverColor sets the menu hoverover color."), Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(WebColorConverter))]
public Color MenuHoverOverColor;
```

Visual Basic

```
<Description("HoverOverColor sets the menu hoverover color."), Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(WebColorConverter))>
Public Property MenuHoverOverColor() As Color
```

Description

`HoverOverColor` sets the menu hoverover color.

DhtmlMenu.MenuStyle Property

Specify the custom style to use. If specified, all specified appearance properties will be ignored.

C#

```
[Description("Specify the custom style to use. If specified, all specified appearance
properties will be ignored."), Category("Ektron")]
public string MenuStyle;
```

Visual Basic

```
<Description("Specify the custom style to use. If specified, all specified appearance
properties will be ignored."), Category("Ektron")>
Public Property MenuStyle() As string
```

DhtmlMenu.Stylesheet Property

Specify the style sheet relative to site root. Leave blank for default.

C#

```
[Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(StringConverter))]
public override string Stylesheet;
```

Visual Basic

```
<Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(StringConverter))>
Public override Property Stylesheet() As string
```

DhtmlMenu.SubMenuBackColor Property

C#

```
[Description("SubMenuBackColor sets the sub menu background color."), Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(WebColorConverter))]
public Color SubMenuBackColor;
```

Visual Basic

```
<Description("SubMenuBackColor sets the sub menu background color."), Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(WebColorConverter))>
Public Property SubMenuBackColor() As Color
```

Description

`SubMenuBackColor` sets the sub menu background color.

DhtmlMenu.SubMenuHoverOverColor Property

C#

```
[Description("SubMenuHoverOverColor sets the sub menu hoverover color."), Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(WebColorConverter))]
public Color SubMenuHoverOverColor;
```

Visual Basic

```
<Description("SubMenuHoverOverColor sets the sub menu hoverover color."), Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(WebColorConverter))>
Public Property SubMenuHoverOverColor() As Color
```

Description

`SubMenuHoverOverColor` sets the menu hoverover color.

DhtmlMenu.SubMenuItemHeight Property

C#

```
[Description("SubMenuItemHeight is the height in pixels of submenu items."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(Int16Converter))]
public Int16 SubMenuItemHeight;
```

Visual Basic

```
<Description("SubMenuItemHeight is the height in pixels of submenu items."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(Int16Converter))>
Public Property SubMenuItemHeight() As Int16
```

Description

`SubMenuItemHeight` is the height in pixels of submenu items.

DhtmlMenu.SubMenuItemWidth Property

C#

```
[Description("SubMenuItemWidth is the width in pixels of submenu items."), Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(Int16Converter))]
public Int16 SubMenuItemWidth;
```

Visual Basic

```
<Description("SubMenuItemWidth is the width in pixels of submenu items."), Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(Int16Converter))>
Public Property SubMenuItemWidth() As Int16
```

Description

`SubMenuItemWidth` is the width in pixels of submenu items.

DhtmlMenu.SubMenuTopOffset Property

C#

```
[Description("SubMenuTopOffset sets the horizontal offset of submenu popups relative to the
parent submenu item."), Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(Int16Converter))]
public Int16 SubMenuTopOffset;
```

Visual Basic

```
<Description("SubMenuTopOffset sets the horizontal offset of submenu popups relative to the
parent submenu item."), Category("Ektron")>
```

```
<Bindable(true)>
<TypeConverter(typeof(Int16Converter))>
Public Property SubMenuTopOffset() As Int16
```

Description

`SubMenuTopOffset` sets the horizontal offset of submenu popups relative to the parent submenu item.

DhtmlMenu.TabIndex Property

Ignored. The property has no effect.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override short TabIndex;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property TabIndex() As short
```

DhtmlMenu.ToolTip Property

Ignored. The property has no effect.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override string ToolTip;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property ToolTip() As string
```

DhtmlMenu.Width Property

The width (in pixels or percent) of the editor, including toolbars.

C#

```
[NotifyParentProperty(true), Browsable(true), Bindable(true), Category("Layout"),
DefaultValue(m_DefaultWidth)]
[Description("The width (in pixels or percent) of the editor, including toolbars.")]
public override Unit Width;
```

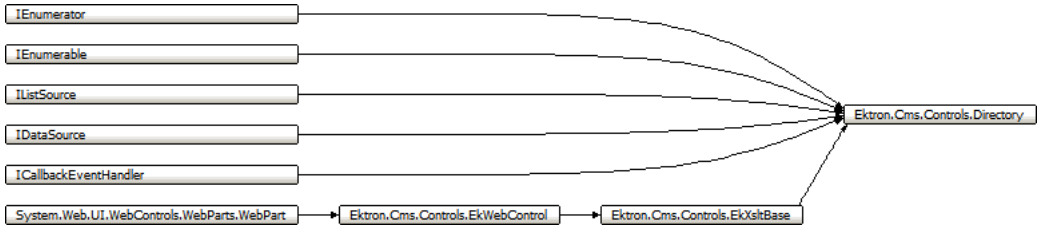
Visual Basic

```
<NotifyParentProperty(true), Browsable(true), Bindable(true), Category("Layout"),
DefaultValue(m_DefaultWidth)>
<Description("The width (in pixels or percent) of the editor, including toolbars.")>
Public override Property Width() As Unit
```

Directory Class

The Directory server control lets you customize the behavior of the Taxonomy (see page 876) feature. You place this control on any Web form to display a taxonomy.

Class Hierarchy



C#

```
[DefaultProperty("Text"), ToolboxData("<{0}:Directory runat=server></{0}:Directory>")]
public class Directory : EkXsltBase, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("Text"), ToolboxData("<{0}:Directory runat=server></{0}:Directory>")>
Public Class Directory
Inherits EkXsltBase
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
Inherits ICallbackEventHandler
```

File

Taxonomy.cs

Directory Methods

The methods of the Directory class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 197)	Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.
	Fill (see page 197)	Populates the data for the Taxonomy (see page 876).
	GetEnumerator (see page 198)	Gets a System.Collections.IEnumerator for the Taxonomy (see page 876).
	GetView (see page 198)	Gets the view data using dataset
	GetViewNames (see page 198)	gets the view names
	MoveNext (see page 198)	Move to next index.
	Reset (see page 199)	Resets the index.
	this (see page 199)	Returns the current ContentItem.ContentRow.
	ToString (see page 199)	Overrides the system ToString function.
	Wrap (see page 199)	Wrap the Html toolbar.

Directory.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

C#

```
public void CopyTo(Array array, int index);
```

Visual Basic

```
Public Function CopyTo(array As Array, index As Integer) As void
```

Directory.Fill Method

Populates the data for the Taxonomy (see page 876).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Directory.GetEnumerator Method

Gets a System.Collections.IEnumerator for the Taxonomy (see page 876).

C#

```
[Browsable(false)]
public IEnumerator GetEnumerator();
```

Visual Basic

```
<Browsable(false)>
Public Function GetEnumerator() As IEnumerator
```

Directory.GetView Method

Gets the view data using dataset

C#

```
public DataSourceView GetView(string viewName);
```

Visual Basic

```
Public Function GetView(viewName As string) As DataSourceView
```

Parameters

Parameters	Description
string viewName	

Returns

DataSourceView

Directory.GetViewNames Method

gets the view names

C#

```
public ICollection GetViewNames();
```

Visual Basic

```
Public Function GetViewNames() As ICollection
```

Returns

View names

Directory.MoveNext Method

Move to next index.

C#

```
public bool MoveNext();
```

Visual Basic

```
Public Function MoveNext() As bool
```

Returns

Next Index

Directory.Reset Method

Resets the index.

C#

```
public void Reset();
```

Visual Basic

```
Public Function Reset() As void
```

Directory.this Indexer

Returns the current ContentItem.ContentRow.

C#

```
[Browsable(false)]
public Ektron.Cms.Controls.ContentItem.ContentRow this[int i];
```

Visual Basic

```
<Browsable(false)>
Public Sub this(i As Integer)
```

Parameters

Parameters	Description
int i	Index value

Returns

Content (see page 822) Row

Directory.ToString Method

Overrides the system ToString function.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Returns

String

Directory.Wrap Method

Wrap the Html toolbar.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	

Directory Properties

The properties of the Directory class are listed here.

Public Properties

	Name	Description
	AddItemFolderId (see page 201)	Set or Get and Folder (see page 683) Id where new items will be added.
	CacheInterval (see page 201)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	Count (see page 202)	Indicates the number of elements in your Taxonomy (see page 876).
	Current (see page 202)	Gets the current index.
	DisplayXslt (see page 202)	The Xslt to use to render the control (optional).
	DynamicParameter (see page 202)	Gets or Sets the QueryString parameter to read a Taxonomy (see page 876) ID dynamically. Set to String.Empty if you wish to always display the default Taxonomy (see page 876) ID (static.)
	EkItems (see page 203)	Returns the array of ContentBase.
	EnableAjax (see page 203)	Set to true to enable Ajax searches. When enabled, the MaxResults (see page 205) property determines the maximum number of results per page. • **True** = Enable Ajax Search (see page 474) (default value) • **False** = Original HTML Search (see page 474)
	EnablePaging (see page 203)	This property, in conjunction with the MaxResults (see page 205) property, lets site visitors view an unlimited number of taxonomy items while controlling the amount of screen space. To accomplish this, the taxonomy display is limited to the number set in the MaxResults (see page 205) property.
	EnableSearch (see page 204)	Set to true if you want a search box (shown below) to appear above this taxonomy display.
	IncludeIcons (see page 204)	Determines whether or not to display an icon next to your navigation lists.
	IsSynchronized (see page 204)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	ItemSortOrder (see page 204)	Specify the sort order of results. Choices are: • **taxonomy_item _display_order** - the order of taxonomy items as set in the Workarea. For additional information, see the CMS400.NET Administrator section "Taxonomy (see page 876)" > "Additional Taxonomy (see page 876)/Category Tasks" > "Change Order of Content (see page 822) Assigned to a Taxonomy (see page 876)/Category" • **content_title** - the content is listed in alphabetical order by title. • **date_created** - content is listed in the order by which it was created. • **last_edit_date** - content is listed in order by its last edit date.
	LinkTarget (see page 205)	Defines the way a link acts when it is clicked. The choices are: • **_blank** = loads the link in a new blank window. This window is not named. • **_self** = loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target. • **_parent** = loads the link in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** = loads the link in the full body of the window. This defaults to acting like "_self" if the document is already... more (see page 205)
	MarkupLanguage (see page 205)	Enter name of a markup file that manages the display of this server control.
	MaxResults (see page 205)	Maximum number of results to be returned. If set to zero, there is no limit.
	Recursive (see page 206)	Determines if subfolders of the selected folders should be brought into the list.
	ShowAllChecked (see page 206)	When set to True a checkmark appears by default in the **show all** check box. This check box shows all categories even when they have no content assigned to them. The **show all** check box is available when a user is logged in.

	ShowCount (see page 206)	Indicates if the number of taxonomy items appears next to each category when displayed in the Web site. The default is: False
	ShowRoot (see page 207)	Determines whether or not to show the actual root name instead of top
	SortDirection (see page 207)	Set the direction of the output
	SuppressWrapperTags (see page 207)	Suppresses the output of span/div tags around the control. As of release 7.5.2, this property is hidden by default. This property is set to false because Ajax uses tags to rewrite the region around the tag. You cannot change the value to **True**.
	SyncRoot (see page 208)	Gets an object that can be used to synchronize the Taxonomy (see page 876).
	TaxonomyCols (see page 208)	Enter the number of columns in which this taxonomy/category will appear on the page.
V	TaxonomyDepth (see page 208)	Enter the number of taxonomy levels to appear below each taxonomy/category. For example, if the taxonomy is Businesses > Restaurants > Pizza and you set TaxonomyDepth to 2, only Business and Restaurants appear. **Important!** - For a live site, Ektron strongly recommends leaving this value at 1. Increasing this value can slow down your live Web server. However, for testing on a staging server, you can increase the depth. A depth greater than 1 is only useful if you create a custom output using the Taxonomy (see page 876)'s XML.
	TaxonomyDescription (see page 208)	Returns taxonomy's description.
V	TaxonomyId (see page 209)	Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.
	TaxonomyItemCols (see page 209)	Enter the number of columns in which this taxonomy/category items(articles) will appear on the page.
	TaxonomyParamId (see page 209)	get or set the dynamic parameter value when dynamic param cannot be passed as url query string.
	TaxonomyPath (see page 209)	Returns taxonomy's path.
	TaxonomyTitle (see page 210)	Returns the Taxonomy (see page 876)'s title.
	TaxonomyTreeData (see page 210)	Returns taxonomy's data.
	TaxonomyXml (see page 210)	Returns taxonomy's XML. (includes subcategory and items)
	XmlDoc (see page 210)	Returns the XML object loaded for Taxonomy (see page 876).

Directory.AddItemFolderId Property

Set or Get and Folder (see page 683) Id where new items will be added.

C#

```
[Browsable(true)]
[Bindable(true)]
[Description("Set or Get and Folder Id where new items will be added.")]
[DefaultValue(0)]
[Category("Ektron")]
public virtual long AddItemFolderId;
```

Visual Basic

```
<Browsable(true)>
<Bindable(true)>
<Description("Set or Get and Folder Id where new items will be added.")>
<DefaultValue(0)>
<Category("Ektron")>
Public virtual Property AddItemFolderId() As long
```

Directory.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true), DefaultValue(0), Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
```

```
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[Personalizable(), WebBrowsable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true), DefaultValue(0), Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<Personalizable(), WebBrowsable()>
Public virtual Property CacheInterval() As double
```

Directory.Count Property

Indicates the number of elements in your Taxonomy (see page 876).

C#

```
[Browsable(false)]
public int Count;
```

Visual Basic

```
<Browsable(false)>
Public Property Count() As Integer
```

Directory.Current Property

Gets the current index.

C#

```
[Browsable(false)]
public object Current;
```

Visual Basic

```
<Browsable(false)>
Public Property Current() As object
```

Directory.DisplayXslt Property

The Xslt to use to render the control (optional).

C#

```
[Bindable(true)]
[Category("Ektron"), DefaultValue("")]
[TypeConverter(typeof(String))]
[Description("The Xslt to use to render the control (optional)")]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron"), DefaultValue("")>
<TypeConverter(typeof(String))>
<Description("The Xslt to use to render the control (optional)")>
Public override Property DisplayXslt() As string
```

Directory.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Taxonomy (see page 876) ID dynamically. Set to String.Empty if you wish to always display the default Taxonomy (see page 876) ID (static.)

C#

```
[Bindable(true), Category("Ektron")]
```

```
[Description("Gets or Sets the QueryString parameter to read a Content ID dynamically - Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true), Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Content ID dynamically - Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

Directory.EkItems Property

Returns the array of ContentBase.

C#

```
[Browsable(false)]
public Ektron.Cms.Common.ContentBase EkItems;
```

Visual Basic

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

Directory.EnableAjax Property

Set to true to enable Ajax searches. When enabled, the MaxResults (see page 205) property determines the maximum number of results per page.

- **True** = Enable Ajax Search (see page 474) (default value)
- **False** = Original HTML Search (see page 474)

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true), Description("Enable/Disable ajax
based retrival.")]
[Personalizable(), WebBrowsable()]
public virtual bool EnableAjax;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true), Description("Enable/Disable ajax
based retrival.")>
<Personalizable(), WebBrowsable()>
Public virtual Property EnableAjax() As bool
```

Directory.EnablePaging Property

This property, in conjunction with the MaxResults (see page 205) property, lets site visitors view an unlimited number of taxonomy items while controlling the amount of screen space. To accomplish this, the taxonomy display is limited to the number set in the MaxResults (see page 205) property.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(0)]
[Description("Set to True to enable paging if the total records exeeds the MaxResult value.
Use this property with non zero value of MaxResults.")]
[Personalizable(), WebBrowsable()]
public bool EnablePaging;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(0)>
<Description("Set to True to enable paging if the total records exeeds the MaxResult value.
```

```
Use this property with non zero value of MaxResults.")>
<Personalizable(), WebBrowsable()>
Public Property EnablePaging() As bool
```

Directory.EnableSearch Property

Set to true if you want a search box (shown below) to appear above this taxonomy display.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true), Browsable(true)]
[Description("When set to True, this property displays a search box to search thru the
taxonomy.")]
[Personalizable(), WebBrowsable()]
public virtual Boolean EnableSearch;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true), Browsable(true)>
<Description("When set to True, this property displays a search box to search thru the
taxonomy.")>
<Personalizable(), WebBrowsable()>
Public virtual Property EnableSearch() As Boolean
```

Directory.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists.

C#

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.")]
[DefaultValue(false)]
[Category("Ektron")]
[Personalizable(), WebBrowsable()]
public bool IncludeIcons;
```

Visual Basic

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.")>
<DefaultValue(false)>
<Category("Ektron")>
<Personalizable(), WebBrowsable()>
Public Property IncludeIcons() As bool
```

Directory.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

```
[Browsable(false)]
public bool IsSynchronized;
```

Visual Basic

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

Directory.ItemSortOrder Property

Specify the sort order of results. Choices are:

- **taxonomy_item _display_order** - the order of taxonomy items as set in the Workarea. For additional information, see the CMS400.NET Administrator section “Taxonomy (see page 876)” > “Additional Taxonomy (see page 876)/Category Tasks” > “Change Order of Content (see page 822) Assigned to a Taxonomy (see page 876)/Category”

- **content_title** - the content is listed in alphabetical order by title.
- **date_created** - content is listed in the order by which it was created.
- **last_edit_date** - content is listed in order by its last edit date.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(TaxonomyItemSortOrder.content_title),
Description("Set the direction of the output.")]
[Personalizable(), WebBrowsable()]
public TaxonomyItemSortOrder ItemSortOrder;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(TaxonomyItemSortOrder.content_title),
Description("Set the direction of the output.")>
<Personalizable(), WebBrowsable()>
Public Property ItemSortOrder() As TaxonomyItemSortOrder
```

Directory.LinkTarget Property

Defines the way a link acts when it is clicked. The choices are:

- **_blank** = loads the link in a new blank window. This window is not named.
- **_self** = loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.
- **_parent** = loads the link in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.
- **_top** = loads the link in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.")]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.")>
Public Property LinkTarget() As ItemLinkTargets
```

Directory.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

Directory.MaxResults Property

Maximum number of results to be returned. If set to zero, there is no limit.

C#

```
[Bindable(true)]
```

```
[Category("Ektron")]
[DefaultValue(0)]
[Description("The maximum number of results to be returned. If set to zero, there is no
limit.")]
[Personalizable(), WebBrowsable()]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The maximum number of results to be returned. If set to zero, there is no
limit.")>
<Personalizable(), WebBrowsable()>
Public Property MaxResults() As Integer
```

Directory.Recursive Property

Determines if subfolders of the selected folders should be brought into the list.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set Recursive to true if you wish to grab results in subfolders of your
FolderPath.")]
public virtual bool Recursive;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set Recursive to true if you wish to grab results in subfolders of your
FolderPath.")>
Public virtual Property Recursive() As bool
```

Directory.ShowAllChecked Property

When set to True a checkmark appears by default in the **show all** check box. This check box shows all categories even when they have no content assigned to them. The **show all** check box is available when a user is logged in.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("false"), Browsable(true)]
[Description("When set to True a checkmark appears by default in the \"show all\" check box.")]
public virtual bool ShowAllChecked;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("false"), Browsable(true)>
<Description("When set to True a checkmark appears by default in the \"show all\" check box.")>
Public virtual Property ShowAllChecked() As bool
```

Directory.ShowCount Property

Indicates if the number of taxonomy items appears next to each category when displayed in the Web site. The default is: False

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(false)]
[Description("Indicates if the number of taxonomy items appears next to each category when
displayed in the Web site. The default is: False")]
[Personalizable(), WebBrowsable()]
public bool ShowCount;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(false)>
<Description("Indicates if the number of taxonomy items appears next to each category when
displayed in the Web site. The default is: False")>
<Personalizable(), WebBrowsable()>
Public Property ShowCount() As bool
```

Directory.ShowRoot Property

Determines whether or not to show the actual root name instead of top

C#

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to show the actual root name instead of top")]
[DefaultValue(false)]
[Category("Ektron")]
[Personalizable(), WebBrowsable()]
public bool ShowRoot;
```

Visual Basic

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to show the actual root name instead of top")>
<DefaultValue(false)>
<Category("Ektron")>
<Personalizable(), WebBrowsable()>
Public Property ShowRoot() As bool
```

Directory.SortDirection Property

Set the direction of the output

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(OrderByDirection.Ascending),
Description("Set the direction of the output.")]
[Personalizable(), WebBrowsable()]
public OrderByDirection SortDirection;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(OrderByDirection.Ascending),
Description("Set the direction of the output.")>
<Personalizable(), WebBrowsable()>
Public Property SortDirection() As OrderByDirection
```

Directory.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control. As of release 7.5.2, this property is hidden by default. This property is set to false because Ajax uses tags to rewrite the region around the tag. You cannot change the value to **True**.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Browsable(false)]
[Description("Suppresses the output of span/div tags around the control. As of release 7.5.2,
this property is hidden by default. This property is set to false because Ajax uses <div> tags
to rewrite the region around the tag. You cannot change the value to <b>True</b>.")]
public override bool SuppressWrapperTags;
```

Visual Basic

```
<Bindable(true)>
```

```
<Category("Ektron")>
<DefaultValue(false)>
<Browsable(false)>
<Description("Suppresses the output of span/div tags around the control. As of release 7.5.2,
this property is hidden by default. This property is set to false because Ajax uses <div> tags
to rewrite the region around the tag. You cannot change the value to <b>True</b>.")>
Public override Property SuppressWrapperTags() As bool
```

Directory.SyncRoot Property

Gets an object that can be used to synchronize the Taxonomy (see page 876).

C#

```
[Browsable(false)]
public object SyncRoot;
```

Visual Basic

```
<Browsable(false)>
Public Property SyncRoot() As object
```

Directory.TaxonomyCols Property

Enter the number of columns in which this taxonomy/category will appear on the page.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)]
[Description("This aligns the taxonomy categories in number of columns.")]
[Personalizable(), WebBrowsable()]
public int TaxonomyCols;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)>
<Description("This aligns the taxonomy categories in number of columns.")>
<Personalizable(), WebBrowsable()>
Public Property TaxonomyCols() As Integer
```

Directory.TaxonomyDepth Property

Enter the number of taxonomy levels to appear below each taxonomy/category. For example, if the taxonomy is Businesses > Restaurants > Pizza and you set TaxonomyDepth to 2, only Business and Restaurants appear.

Important! - For a live site, Ektron strongly recommends leaving this value at 1. Increasing this value can slow down your live Web server. However, for testing on a staging server, you can increase the depth.

A depth greater than 1 is only useful if you create a custom output using the Taxonomy (see page 876)'s XML.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)]
[Description("Enter the number of taxonomy levels to appear below each taxonomy/category. For
example, if the taxonomy is Businesses > Restaurants > Pizza and you set TaxonomyDepth to 2,
only Business and Restaurants appear.")]
public virtual int TaxonomyDepth;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)>
<Description("Enter the number of taxonomy levels to appear below each taxonomy/category. For
example, if the taxonomy is Businesses > Restaurants > Pizza and you set TaxonomyDepth to 2,
only Business and Restaurants appear.")>
Public virtual Property TaxonomyDepth() As Integer
```

Directory.TaxonomyDescription Property

Returns taxonomy's description.

C#

```
public string TaxonomyDescription;
```

Visual Basic

```
Public Property TaxonomyDescription() As string
```

Directory.TaxonomyId Property

Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)]
[Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")]
[Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public virtual long TaxonomyId;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)>
<Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")>
<Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public virtual Property TaxonomyId() As long
```

Directory.TaxonomyItemCols Property

Enter the number of columns in which this taxonomy/category items(articles) will appear on the page.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)]
[Description("This aligns the taxonomy categories in number of columns.")]
[Personalizable(), WebBrowsable()]
public int TaxonomyItemCols;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)>
<Description("This aligns the taxonomy categories in number of columns.")>
<Personalizable(), WebBrowsable()>
Public Property TaxonomyItemCols() As Integer
```

Directory.TaxonomyParamId Property

get or set the dynamic parameter value when dynamic param cannot be passed as url query string.

C#

```
public long TaxonomyParamId;
```

Visual Basic

```
Public Property TaxonomyParamId() As long
```

Directory.TaxonomyPath Property

Returns taxonomy's path.

C#

```
public string TaxonomyPath;
```

Visual Basic

```
Public Property TaxonomyPath() As string
```

Directory.TaxonomyTitle Property

Returns the Taxonomy (see page 876)'s title.

C#

```
public string TaxonomyTitle;
```

Visual Basic

```
Public Property TaxonomyTitle() As string
```

Directory.TaxonomyTreeData Property

Returns taxonomy's data.

C#

```
public TaxonomyData TaxonomyTreeData;
```

Visual Basic

```
Public Property TaxonomyTreeData() As TaxonomyData
```

Directory.TaxonomyXml Property

Returns taxonomy's XML. (includes subcategory and items)

C#

```
public string TaxonomyXml;
```

Visual Basic

```
Public Property TaxonomyXml() As string
```

Directory.XmlDoc Property

Returns the XML object loaded for Taxonomy (see page 876).

C#

```
public override XmlDocument XmlDoc;
```

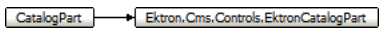
Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

EktronCatalogPart Class

When placed inside a Microsoft CatalogZone, the EktronCatalogPart allows users to choose the type of CMS400.NET control that is added to a WebPartZone.

Class Hierarchy



C#

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:EktronCatalogPart runat=server></{0}:EktronCatalogPart>")]
public class EktronCatalogPart : CatalogPart;
```

Visual Basic

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:EktronCatalogPart runat=server></{0}:EktronCatalogPart>")>
Public Class EktronCatalogPart
Inherits CatalogPart
```

File

CatalogPart.cs

EktronCatalogPart Properties

The properties of the EktronCatalogPart class are listed here.

Public Properties

	Name	Description
	BlogID (see page 211)	The ID of the blog in CMS400.NET.
	BoardID (see page 212)	The ID of the discussion board.
	CalendarDynamicParameter (see page 212)	To make this calendar dynamic, select calendar_id. When you do, this server control uses the calendar passed as a URL parameter.
	CollectionDynamicParameter (see page 212)	To make this collection dynamic, select id. When you do, this server control uses the collection passed as a URL parameter.
	ContentBlockDynamicParameter (see page 213)	Select id. When you do, this server control uses the content block passed as a URL parameter.
	DefaultCalendarID (see page 213)	The ID number of the calendar that appears where you inserted this server control.
	DefaultCollectionID (see page 213)	The ID of a collection that appears where you inserted this server control if no other collection is identified, or is not available.
	DefaultContentID (see page 213)	The ID of a content block that appears where you insert this server control.
	DefaultFormID (see page 214)	The ID of a FormBlock (see page 254) that appears where you inserted this server control if no other form block is identified.
	EnableBlogControl (see page 214)	Set to True to allow a Blog (see page 32) Control to appear in the catalog.
	EnableCalendarControl (see page 214)	Set to true to allow a Calendar (see page 76) control to appear in the Catalog.
	EnableCollectionControl (see page 215)	Set to True to allow a Collection (see page 95) control to appear in the Catalog.
	EnableContentBlockControl (see page 215)	Set to True to allow a Content (see page 822) Block Control to appear in the catalog.
	EnableDiscussionBoardsControl (see page 215)	Set to True to allow a Forum (see page 264) (Discussion Board) control to appear in the catalog.
	EnableFormBlockControl (see page 215)	Set to True to allow a FormBlock (see page 254) Control to appear in the catalog.
	EnableListSummaryControl (see page 216)	Set to True to allow a ListSummary (see page 335) Control to appear in the catalog.
	EnableThreadedDiscussionsControl (see page 216)	Set to True to allow a Forum (see page 264) (Discussion Board) control to appear in the catalog.
	FolderID (see page 216)	The folder that contains the content blocks which appear in the list summary.
	FormBlockDynamicParameter (see page 216)	To make this form block dynamic, select ekfrm. When you do, this server control uses the form block passed as a URL parameter.
	Title (see page 217)	The descriptive name of the Ektron Catalog. The title appears as clickable link in the catalog when the Ektron Catalog is not shown.

EktronCatalogPart.BlogID Property

The ID of the blog in CMS400.NET.

C#

```
[Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Browsable(true), DefaultValue(0)]
[Category("Ektron Blog Control")]
[Description("The ID of the blog in CMS400.NET.")]
public long BlogID;
```

Visual Basic

```
<Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
```

```
<Browsable(true), DefaultValue(0)>
<Category("Ektron Blog Control")>
<Description("The ID of the blog in CMS400.NET.")>
Public Property BlogID() As long
```

EktronCatalogPart.BoardID Property

The ID of the discussion board.

C#

```
[Browsable(true), DefaultValue(0)]
[Category("Ektron Discussion Boards Control")]
[Description("The ID of the discussion board.")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BoardID;
```

Visual Basic

```
<Browsable(true), DefaultValue(0)>
<Category("Ektron Discussion Boards Control")>
<Description("The ID of the discussion board.")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BoardID() As long
```

EktronCatalogPart.CalendarDynamicParameter Property

To make this calendar dynamic, select calendar_id. When you do, this server control uses the calendar passed as a URL parameter.

C#

```
[TypeConverter(typeof(DynamicCalendarTypeConverter))]
[Browsable(true), DefaultValue("")]
[Category("Ektron Calendar Control")]
[Description("To make this calendar dynamic, select calendar_id. When you do, this server
control uses the calendar passed as a URL parameter.")]
public string CalendarDynamicParameter;
```

Visual Basic

```
<TypeConverter(typeof(DynamicCalendarTypeConverter))>
<Browsable(true), DefaultValue("")>
<Category("Ektron Calendar Control")>
<Description("To make this calendar dynamic, select calendar_id. When you do, this server
control uses the calendar passed as a URL parameter.")>
Public Property CalendarDynamicParameter() As string
```

EktronCatalogPart.CollectionDynamicParameter Property

To make this collection dynamic, select id. When you do, this server control uses the collection passed as a URL parameter.

C#

```
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Browsable(true), DefaultValue("")]
[Category("Ektron Collection Control")]
[Description("To make this collection dynamic, select id. When you do, this server control
uses the collection passed as a URL parameter.")]
public string CollectionDynamicParameter;
```

Visual Basic

```
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Browsable(true), DefaultValue("")>
<Category("Ektron Collection Control")>
<Description("To make this collection dynamic, select id. When you do, this server control
uses the collection passed as a URL parameter.")>
Public Property CollectionDynamicParameter() As string
```

EktronCatalogPart.ContentBlockDynamicParameter Property

Select id. When you do, this server control uses the content block passed as a URL parameter.

C#

```
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Browsable(true), DefaultValue("")]
[Category("Ektron Content Block")]
[Description("Select id. When you do, this server control uses the content block passed as a
URL parameter.")]
public string ContentBlockDynamicParameter;
```

Visual Basic

```
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Browsable(true), DefaultValue("")>
<Category("Ektron Content Block")>
<Description("Select id. When you do, this server control uses the content block passed as a
URL parameter.")>
Public Property ContentBlockDynamicParameter() As string
```

EktronCatalogPart.DefaultCalendarID Property

The ID number of the calendar that appears where you inserted this server control.

C#

```
[Editor(typeof(CalendarUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Browsable(true), DefaultValue(0)]
[Category("Ektron Calendar Control")]
[Description("The ID number of the calendar that appears where you inserted this server
control.")]
public long DefaultCalendarID;
```

Visual Basic

```
<Editor(typeof(CalendarUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron Calendar Control")>
<Description("The ID number of the calendar that appears where you inserted this server
control.")>
Public Property DefaultCalendarID() As long
```

EktronCatalogPart.DefaultCollectionID Property

The ID of a collection that appears where you inserted this server control if no other collection is identified, or is not available.

C#

```
[Browsable(true), DefaultValue(0)]
[Editor(typeof(CollectionUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Category("Ektron Collection Control")]
[Description("The ID of a collection that appears where you inserted this server control if no
other collection is identified, or is not available.")]
public long DefaultCollectionID;
```

Visual Basic

```
<Browsable(true), DefaultValue(0)>
<Editor(typeof(CollectionUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Category("Ektron Collection Control")>
<Description("The ID of a collection that appears where you inserted this server control if no
other collection is identified, or is not available.")>
Public Property DefaultCollectionID() As long
```

EktronCatalogPart.DefaultContentID Property

The ID of a content block that appears where you insert this server control.

C#

```
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Browsable(true), DefaultValue(0)]
[Category("Ektron Content Block")]
[Description("The ID of a content block that appears where you insert this server control.")]
public long DefaultContentID;
```

Visual Basic

```
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron Content Block")>
<Description("The ID of a content block that appears where you insert this server control.")>
Public Property DefaultContentID() As long
```

EktronCatalogPart.DefaultFormID Property

The ID of a FormBlock (see page 254) that appears where you inserted this server control if no other form block is identified.

C#

```
[Editor(typeof(FormUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Browsable(true), DefaultValue(0)]
[Category("Ektron FormBlock Control")]
[Description("The ID of a FormBlock that appears where you inserted this server control if no
other form block is identified.")]
public long DefaultFormID;
```

Visual Basic

```
<Editor(typeof(FormUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron FormBlock Control")>
<Description("The ID of a FormBlock that appears where you inserted this server control if no
other form block is identified.")>
Public Property DefaultFormID() As long
```

EktronCatalogPart.EnableBlogControl Property

Set to True to allow a Blog (see page 32) Control to appear in the catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Blog Control")]
[Description("Set to True to allow a Blog Control to appear in the catalog.")]
public bool EnableBlogControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Blog Control")>
<Description("Set to True to allow a Blog Control to appear in the catalog.")>
Public Property EnableBlogControl() As bool
```

EktronCatalogPart.EnableCalendarControl Property

Set to true to allow a Calendar (see page 76) control to appear in the Catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Calendar Control")]
[Description("Set to true to allow a Calendar control to appear in the Catalog.")]
public bool EnableCalendarControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
```

```
<Category("Ektron Calendar Control")>
<Description("Set to true to allow a Calendar control to appear in the Catalog.")>
Public Property EnableCalendarControl() As bool
```

EktronCatalogPart.EnableCollectionControl Property

Set to True to allow a Collection (see page 95) control to appear in the Catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Collection Control")]
[Description("Set to True to allow a Collection control to appear in the Catalog.")]
public bool EnableCollectionControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Collection Control")>
<Description("Set to True to allow a Collection control to appear in the Catalog.")>
Public Property EnableCollectionControl() As bool
```

EktronCatalogPart.EnableContentBlockControl Property

Set to True to allow a Content (see page 822) Block Control to appear in the catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Content Block")]
[Description("Set to True to allow a Content Block Control to appear in the catalog.")]
public bool EnableContentBlockControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Content Block")>
<Description("Set to True to allow a Content Block Control to appear in the catalog.")>
Public Property EnableContentBlockControl() As bool
```

EktronCatalogPart.EnableDiscussionBoardsControl Property

Set to True to allow a Forum (see page 264) (Discussion Board) control to appear in the catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Discussion Boards Control")]
[Description("Set to True to allow a Forum (Discussion Board) control to appear in the
catalog.")]
public bool EnableDiscussionBoardsControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Discussion Boards Control")>
<Description("Set to True to allow a Forum (Discussion Board) control to appear in the
catalog.")>
Public Property EnableDiscussionBoardsControl() As bool
```

EktronCatalogPart.EnableFormBlockControl Property

Set to True to allow a FormBlock (see page 254) Control to appear in the catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron FormBlock Control")]
[Description("Set to True to allow a FormBlock Control to appear in the catalog.")]
public bool EnableFormBlockControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron FormBlock Control")>
<Description("Set to True to allow a FormBlock Control to appear in the catalog.")>
Public Property EnableFormBlockControl() As bool
```

EktronCatalogPart.EnableListSummaryControl Property

Set to True to allow a ListSummary (see page 335) Control to appear in the catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron ListSummary")]
[Description("Set to True to allow a ListSummary Control to appear in the catalog.")]
public bool EnableListSummaryControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron ListSummary")>
<Description("Set to True to allow a ListSummary Control to appear in the catalog.")>
Public Property EnableListSummaryControl() As bool
```

EktronCatalogPart.EnableThreadedDiscussionsControl Property

Set to True to allow a Forum (see page 264) (Discussion Board) control to appear in the catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Discussion Boards Control")]
[Description("Set to True to allow a Forum (Discussion Board) control to appear in the
catalog.")]
public bool EnableThreadedDiscussionsControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Discussion Boards Control")>
<Description("Set to True to allow a Forum (Discussion Board) control to appear in the
catalog.")>
Public Property EnableThreadedDiscussionsControl() As bool
```

EktronCatalogPart.FolderID Property

The folder that contains the content blocks which appear in the list summary.

C#

```
[Editor(typeof(FolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Browsable(true), DefaultValue(0)]
[Category("Ektron ListSummary")]
[Description("The folder that contains the content blocks which appear in the list summary.")]
public long FolderID;
```

Visual Basic

```
<Editor(typeof(FolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron ListSummary")>
<Description("The folder that contains the content blocks which appear in the list summary.")>
Public Property FolderID() As long
```

EktronCatalogPart.FormBlockDynamicParameter Property

To make this form block dynamic, select ekfrm. When you do, this server control uses the form block passed as a URL parameter.

C#

```
[TypeConverter(typeof(DynamicFormIDTypeConverter))]
[Browsable(true), DefaultValue("")]
[Category("Ektron FormBlock Control")]
[Description("To make this form block dynamic, select ekfrm. When you do, this server control
uses the form block passed as a URL parameter.")]
public string FormBlockDynamicParameter;
```

Visual Basic

```
<TypeConverter(typeof(DynamicFormIDTypeConverter))>
<Browsable(true), DefaultValue("")>
<Category("Ektron FormBlock Control")>
<Description("To make this form block dynamic, select ekfrm. When you do, this server control
uses the form block passed as a URL parameter.")>
Public Property FormBlockDynamicParameter() As string
```

EktronCatalogPart.Title Property

The descriptive name of the Ektron Catalog. The title appears as clickable link in the catalog when the Ektron Catalog is not shown.

C#

```
[Browsable(true), DefaultValue("Ektron Catalog Part")]
[Category("Ektron")]
[Description("The descriptive name of the Ektron Catalog. The title appears as clickable link
in the catalog when the Ektron Catalog is not shown.")]
public override string Title;
```

Visual Basic

```
<Browsable(true), DefaultValue("Ektron Catalog Part")>
<Category("Ektron")>
<Description("The descriptive name of the Ektron Catalog. The title appears as clickable link
in the catalog when the Ektron Catalog is not shown.")>
Public override Property Title() As string
```

EktronCommunityCatalogPart Class

The EktronCommunityCatalogPart Server Control gets a list of Discussion Board (Forums) and Blogs from your site. Then, it allows site users to choose from any available Discussion Boards and Blogs and add them to a WebPartZone.

Class Hierarchy

C#

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:EktronCommunityCatalogPart runat=server></{0}:EktronCommunityCatalogPart>")]
public class EktronCommunityCatalogPart : CatalogPart;
```

Visual Basic

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:EktronCommunityCatalogPart runat=server></{0}:EktronCommunityCatalogPart>")>
Public Class EktronCommunityCatalogPart
Inherits CatalogPart
```

File

CommunityCatalogPart.cs

EktronCommunityCatalogPart Properties

The properties of the EktronCommunityCatalogPart class are listed here.

Public Properties

	Name	Description
	EnableBlogs (see page 218)	When set to True, the option to add a blog control to a Web part is available.
	EnableDiscussionBoards (see page 218)	When set to True, the option to add a Forum (see page 264) (Discussion Board) to a Web part is available.
	EnableForums (see page 218)	When set to True, the option to add a Forum (see page 264) (Discussion Board) to a Web part is available.
	Title (see page 219)	The descriptive name of the Ektron Community (see page 750) Catalog. The title appears as clickable link in the catalog when the Ektron Catalog is not shown. The default title is Ektron Community (see page 750) Catalog Part.

EktronCommunityCatalogPart.EnableBlogs Property

When set to True, the option to add a blog control to a Web part is available.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron")]
[Description("When set to True, the option to add a blog control to a Web part is available.")]
public bool EnableBlogs;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron")>
<Description("When set to True, the option to add a blog control to a Web part is available.")>
Public Property EnableBlogs() As bool
```

EktronCommunityCatalogPart.EnableDiscussionBoards Property

When set to True, the option to add a Forum (see page 264) (Discussion Board) to a Web part is available.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron")]
[Description("When set to True, the option to add a Forum (Discussion Board) to a Web part is
available.")]
public bool EnableDiscussionBoards;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron")>
<Description("When set to True, the option to add a Forum (Discussion Board) to a Web part is
available.")>
Public Property EnableDiscussionBoards() As bool
```

EktronCommunityCatalogPart.EnableForums Property

When set to True, the option to add a Forum (see page 264) (Discussion Board) to a Web part is available.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron")]
[Description("When set to True, the option to add a Forum (Discussion Board) to a Web part is
available.")]
public bool EnableForums;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron")>
<Description("When set to True, the option to add a Forum (Discussion Board) to a Web part is
available.")>
Public Property EnableForums() As bool
```

EktronCommunityCatalogPart.Title Property

The descriptive name of the Ektron Community (see page 750) Catalog. The title appears as clickable link in the catalog when the Ektron Catalog is not shown. The default title is Ektron Community (see page 750) Catalog Part.

C#

```
[Browsable(true), DefaultValue("Ektron Catalog Part")]
[Category("Ektron")]
[Description("The descriptive name of the Ektron Community Catalog. The title appears as
clickable link in the catalog when the Ektron Catalog is not shown. The default title is
Ektron Community Catalog Part.")]
public override string Title;
```

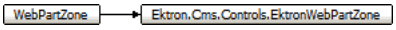
Visual Basic

```
<Browsable(true), DefaultValue("Ektron Catalog Part")>
<Category("Ektron")>
<Description("The descriptive name of the Ektron Community Catalog. The title appears as
clickable link in the catalog when the Ektron Catalog is not shown. The default title is
Ektron Community Catalog Part.")>
Public override Property Title() As string
```

EktronWebPartZone Class

Similar to a Microsoft WebPartZone, the EktronWebPartZone server control provides a location for content to be displayed on a Web page. The difference between the two is formatting. The EktronWebPartZone is designed to work specifically with Ektron Server Controls (see page 7), while Microsoft WebPartZones are more generic.

Class Hierarchy



C#

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:EktronWebPartZone runat=server></{0}:EktronWebPartZone>")]
public class EktronWebPartZone : WebPartZone;
```

Visual Basic

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:EktronWebPartZone runat=server></{0}:EktronWebPartZone>")>
Public Class EktronWebPartZone
Inherits WebPartZone
```

File

EkWebPartZone.cs

EktronWebPartZone Properties

The properties of the EktronWebPartZone class are listed here.

Public Properties

	Name	Description
	CmsSitePath (see page 219)	DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time.

EktronWebPartZone.CmsSitePath Property

DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time.

C#

```
[Browsable(false)]
```

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Editor(typeof(SelectWSUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Description("DESIGN TIME ONLY: Used to find the web service required to access the CMS in
Design Time.")]
public string CmsSitePath;
```

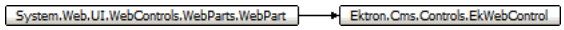
Visual Basic

```
<Browsable(false)>
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Editor(typeof(SelectWSUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("DESIGN TIME ONLY: Used to find the web service required to access the CMS in
Design Time.")>
Public Property CmsSitePath() As string
```

EkWebControl Class

The base class from which all Ektron CMS400 Server Controls (see page 7) are derived.

Class Hierarchy



C#

```
[ToolboxData("<{0}:EkWebControl runat=server></{0}:EkWebControl>")]
public abstract class EkWebControl : System.Web.UI.WebControls.WebParts.WebPart;
```

Visual Basic

```
<ToolboxData("<{0}:EkWebControl runat=server></{0}:EkWebControl>")>
Public abstract Class EkWebControl
Inherits System.Web.UI.WebControls.WebParts.WebPart
```

File

EkWebControl.cs

EkWebControl.EkWebControl Constructor

Constructor for the Ektron Server Control base class - initializes the CmsConnection layer

C#

```
public EkWebControl();
```

Visual Basic

```
Public Sub New()
```

EkWebControl Methods

The methods of the EkWebControl class are listed here.

Public Methods

	Name	Description
V	Fill (see page 221)	Method that can be overridden and used to grab data from the back end for each server control.
S	GetEnumDescription (see page 221)	Gets the Description attribute value for the enum value
S	GetFilenameFromPath (see page 221)	Returns the filename (with extention), given a path and filename.

	GetMessage (see page 222)	This is the overview for the GetMessage method overload.
	LocalizeString (see page 222)	Used to localize strings in the control properties window and at render time. Strings should be defined in resource files. Any property that the user would like to be localized to the sites default language should begin with the constant STR_LOCALIZE defined in the EkWebControl (see page 220) class.
	RedirectFromLoginPage (see page 223)	Provides a mechanism to return from the login page to the previous page (if specified in web.config and on the querystring).
	RedirectToLoginPage (see page 223)	Redirects to the login page (if specified in web.config) and the user is not currently logged in.
	RegisterAllDmsMenuCssAndScripts (see page 223)	Used to register all necessary DmsMenu CSS and Scripts in one shot
	RegisterDmsMenuCSS (see page 223)	Used to register the DmsMenu.css File for the DmsMenu
	RegisterDmsMenuScript (see page 224)	Used to register the DMSMenu.js file
	RegisterJQueryScript (see page 224)	Used to register the Ektron.js file
	RegisterMembershipTabScript (see page 224)	Used to Register the Membership (see page 360) tab JS and CSS files
	RegisterThickBoxCSS (see page 224)	Used to Register the box.css file for ThickBox
	RegisterThickBoxScript (see page 224)	Used to Register the thickbox.js file
	ToString (see page 224)	Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.
V	Wrap (see page 225)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

EkWebControl.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public virtual void Fill();
```

Visual Basic

```
Public virtual Function Fill() As void
```

EkWebControl.GetEnumDescription Method

Gets the Description attribute value for the enum value

C#

```
public static string GetEnumDescription(Enum value);
```

Visual Basic

```
Public static Function GetEnumDescription(value As Enum) As string
```

Parameters

Parameters	Description
Enum value	

EkWebControl.GetFilenameFromPath Method

Returns the filename (with extention), given a path and filename.

C#

```
public static string GetFilenameFromPath(string path);
```

Visual Basic

```
Public static Function GetFilenameFromPath(path As string) As string
```

Parameters

Parameters	Description
string path	

GetMessage Method

This is the overview for the GetMessage method overload.

Overload List

	Name	Description
	EkWebControl.GetMessage (string) (see page 222)	Returns the localized value of the supplied string.
	EkWebControl.GetMessage (string, string) (see page 222)	Returns the localized value of the supplied string, or the default string if it doesn't exist in the resources.

EkWebControl.GetMessage Method (string)

Returns the localized value of the supplied string.

C#

```
public string GetMessage(string resourceString);
```

Visual Basic

```
Public Function GetMessage(resourceString As string) As string
```

Parameters

Parameters	Description
string resourceString	

EkWebControl.GetMessage Method (string, string)

Returns the localized value of the supplied string, or the default string if it doesn't exist in the resources.

C#

```
public string GetMessage(string resourceString, string defaultString);
```

Visual Basic

```
Public Function GetMessage(resourceString As string, defaultString As string) As string
```

Parameters

Parameters	Description
string resourceString	
string defaultString	

EkWebControl.LocalizeString Method

Used to localize strings in the control properties window and at render time. Strings should be defined in resource files. Any property that the user would like to be localized to the sites default language should begin with the constant STR_LOCALIZE defined in the EkWebControl (see page 220) class.

C#

```
public string LocalizeString(string strInputText);
```

Visual Basic

```
Public Function LocalizeString(strInputText As string) As string
```

Parameters

Parameters	Description
string strInputText	String to be localized

Returns

Input string if it does not begin with STR_LOCALIZE. Localized string if successfully localized using lower level API (see page 669).

EkWebControl.RedirectFromLoginPage Method

Provides a mechanism to return from the login page to the previous page (if specified in web.config and on the querystring).

C#

```
public bool RedirectFromLoginPage();
```

Visual Basic

```
Public Function RedirectFromLoginPage() As bool
```

Returns

bool

EkWebControl.RedirectToLoginPage Method

Redirects to the login page (if specified in web.config) and the user is not currently logged in.

C#

```
public bool RedirectToLoginPage();
```

Visual Basic

```
Public Function RedirectToLoginPage() As bool
```

Returns

bool

EkWebControl.RegisterAllDmsMenuCssAndScripts Method

Used to register all necessary DmsMenu CSS and Scripts in one shot

C#

```
public void RegisterAllDmsMenuCssAndScripts();
```

Visual Basic

```
Public Function RegisterAllDmsMenuCssAndScripts() As void
```

EkWebControl.RegisterDmsMenuCSS Method

Used to register the DmsMenu.css File for the DmsMenu

C#

```
public void RegisterDmsMenuCSS();
```

Visual Basic

```
Public Function RegisterDmsMenuCSS() As void
```

EkWebControl.RegisterDmsMenuScript Method

Used to register the DMSMenu.js file

C#

```
public void RegisterDmsMenuScript();
```

Visual Basic

```
Public Function RegisterDmsMenuScript() As void
```

EkWebControl.RegisterJQueryScript Method

Used to register the Ektron.js file

C#

```
public void RegisterJQueryScript();
```

Visual Basic

```
Public Function RegisterJQueryScript() As void
```

EkWebControl.RegisterMembershipTabScript Method

Used to Register the Membership (see page 360) tab JS and CSS files

C#

```
public void RegisterMembershipTabScript();
```

Visual Basic

```
Public Function RegisterMembershipTabScript() As void
```

EkWebControl.RegisterThickBoxCSS Method

Used to Register the box.css file for ThickBox

C#

```
public void RegisterThickBoxCSS();
```

Visual Basic

```
Public Function RegisterThickBoxCSS() As void
```

EkWebControl.RegisterThickBoxScript Method

Used to Register the thickbox.js file

C#

```
public void RegisterThickBoxScript();
```

Visual Basic

```
Public Function RegisterThickBoxScript() As void
```

EkWebControl.ToString Method

Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

EkWebControl.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public new virtual string Wrap(string Html);
```

Visual Basic

```
Public new virtual Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

EkWebControl Properties

The properties of the EkWebControl class are listed here.

Public Properties

	Name	Description
	Authenticated (see page 225)	DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and permits them to login through a login dialog in Visual Studio.NET.
	CmsSitePath (see page 226)	DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time.
	Error (see page 226)	An exception caught in the rendering of this Server Control.
	Hide (see page 226)	Alias for the 'Visible' property. Visible is set to mirror this value. This property is used to hide output in design time, whereas Visible does not hide output in design time.
	IsLoggedIn (see page 227)	DESIGN TIME ONLY: Read only property that lets a developer know if the user is currently log into the site as a user.
	Language (see page 227)	Set a language for viewing content blocks, collections, etc. in different languages. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).
	LanguageID (see page 227)	This property lets a designer view content blocks, collections etc in different languages.
	loggedInUserID (see page 227)	DESIGN TIME ONLY: Read only property that lets a developer know the user ID of the user who is currently logged in.
	loggedInUserName (see page 228)	DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currently logged in.
	LogLevel (see page 228)	"Allows control-level filtering of the logged events, from verbose to none
	SuppressWrapperTags (see page 228)	Suppresses the output of span/div tags around the control.
	Text (see page 228)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.
	WrapTag (see page 229)	Allows a developer to specify control's tag. Default is assumed "".

EkWebControl.Authenticated Property

DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and permits them to login through a login dialog in Visual Studio.NET.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(LoginUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(false)]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Description("DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and
permits them to login through a login dialog in VisualStudio.NET.")]
public string Authenticated;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(LoginUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(false)>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Description("DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and
permits them to login through a login dialog in VisualStudio.NET.")>
Public Property Authenticated() As string
```

EkWebControl.CmsSitePath Property

DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time.

C#

```
[Browsable(false)]
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Editor(typeof(SelectWSUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Description("DESIGN TIME ONLY: Used to find the web service required to access the CMS in
Design Time.")]
public string CmsSitePath;
```

Visual Basic

```
<Browsable(false)>
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Editor(typeof(SelectWSUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("DESIGN TIME ONLY: Used to find the web service required to access the CMS in
Design Time.")>
Public Property CmsSitePath() As string
```

EkWebControl.Error Property

An exception caught in the rendering of this Server Control.

C#

```
[Bindable(true)]
[DefaultValue("")]
[Description("An exception caught in the rendering of this Server Control.")]
public string Error;
```

Visual Basic

```
<Bindable(true)>
<DefaultValue("")>
<Description("An exception caught in the rendering of this Server Control.")>
Public Property Error() As string
```

EkWebControl.Hide Property

Alias for the 'Visible' property. Visible is set to mirror this value. This property is used to hide output in design time, whereas Visible does not hide output in design time.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Alias for the 'Visible' property. Visible is set to mirror this value. This
property is used to hide output in design time, whereas Visible does not hide output in design
```

```
time.")]
public bool Hide;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Alias for the 'Visible' property. Visible is set to mirror this value. This
property is used to hide output in design time, whereas Visible does not hide output in design
time.")>
Public Property Hide() As bool
```

EkWebControl.IsLoggedIn Property

DESIGN TIME ONLY: Read only property that lets a developer know if the user is currently log into the site as a user.

C#

```
public bool IsLoggedIn;
```

Visual Basic

```
Public Property IsLoggedIn() As bool
```

EkWebControl.Language Property

Set a language for viewing content blocks, collections, etc. in different languages. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(1033)]
[Description("Set a language for viewing content blocks, collections, etc. in different
languages. The language property shows results in design-time (in Visual Studio) and at
run-time (in a browser).")]
[Editor(typeof(SelectLanguageIDTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int Language;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(1033)>
<Description("Set a language for viewing content blocks, collections, etc. in different
languages. The language property shows results in design-time (in Visual Studio) and at
run-time (in a browser).")>
<Editor(typeof(SelectLanguageIDTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property Language() As Integer
```

EkWebControl.LanguageID Property

This property lets a designer view content blocks, collections etc in different languages.

C#

```
[Browsable(false)]
public int LanguageID;
```

Visual Basic

```
<Browsable(false)>
Public Property LanguageID() As Integer
```

EkWebControl.loggedInUserID Property

DESIGN TIME ONLY: Read only property that lets a developer know the user ID of the user who is currently logged in.

C#

```
public long loggedInUserID;
```

Visual Basic

```
Public Property loggedInUserID() As long
```

EkWebControl.loggedInUserName Property

DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currently logged in.

C#

```
public string loggedInUserName;
```

Visual Basic

```
Public Property loggedInUserName() As string
```

EkWebControl.LogLevel Property

"Allows control-level filtering of the logged events, from verbose to none

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(System.Diagnostics.TraceLevel))]
[Description("Allows control-level filtering of the logged events, from verbose to none")]
[DefaultValue("Error")]
public System.Diagnostics.TraceLevel LogLevel;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(System.Diagnostics.TraceLevel))>
<Description("Allows control-level filtering of the logged events, from verbose to none")>
<DefaultValue("Error")>
Public Property LogLevel() As System.Diagnostics.TraceLevel
```

EkWebControl.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Suppresses the output of span/div tags around the control.")]
public virtual bool SuppressWrapperTags;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Suppresses the output of span/div tags around the control.")>
Public virtual Property SuppressWrapperTags() As bool
```

EkWebControl.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

```
[Bindable(true)]
```

```
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to treat the server control as a Label, taking information
from the Control and painting it on top of the Control for display. Leave blank for default
display.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public virtual string Text;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows a developer to treat the server control as a Label, taking information
from the Control and painting it on top of the Control for display. Leave blank for default
display.")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Browsable(false)>
Public virtual Property Text() As string
```

EkWebControl.WrapTag Property

Allows a developer to specify control's tag. Default is assumed "".

C#

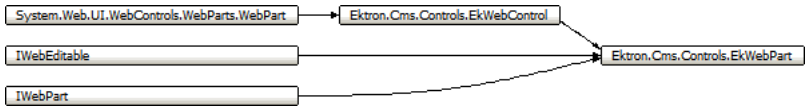
```
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("span")]
[TypeConverter(typeof(WrapTagTypeConverter))]
[Description("Allows a developer to specify control's tag. Default is assumed '<span>'.")]
public virtual string WrapTag;
```

Visual Basic

```
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("span")>
<TypeConverter(typeof(WrapTagTypeConverter))>
<Description("Allows a developer to specify control's tag. Default is assumed '<span>'.")>
Public virtual Property WrapTag() As string
```

EkWebPart Class

Class Hierarchy



C#

```
public abstract class EkWebPart : EkWebControl, IWebEditable, IWebPart;
```

Visual Basic

```
Public abstract Class EkWebPart
Inherits EkWebControl
Inherits IWebEditable
Inherits IWebPart
```

File

EkWebPart.cs

Description

This is class Ektron.Cms.Controls.EkWebPart.

EkWebPart Methods

The methods of the EkWebPart class are listed here.

Public Methods

	Name	Description
	CreateEditorParts (see page 230)	This is CreateEditorParts, a member of class EkWebPart.

EkWebPart.CreateEditorParts Method

C#

```
public EditorPartCollection CreateEditorParts();
```

Visual Basic

```
Public Function CreateEditorParts() As EditorPartCollection
```

Description

This is CreateEditorParts, a member of class EkWebPart.

EkWebPart Properties

The properties of the EkWebPart class are listed here.

Public Properties

	Name	Description
	CatalogIconImageUrl (see page 230)	This is CatalogIconImageUrl, a member of class EkWebPart.
	Description (see page 230)	This is Description, a member of class EkWebPart.
	Subtitle (see page 231)	This is Subtitle, a member of class EkWebPart.
	Title (see page 231)	This is Title, a member of class EkWebPart.
	TitleIconImageUrl (see page 231)	This is TitleIconImageUrl, a member of class EkWebPart.
	TitleUrl (see page 231)	This is TitleUrl, a member of class EkWebPart.
	WebBrowsableObject (see page 231)	This is WebBrowsableObject, a member of class EkWebPart.

EkWebPart.CatalogIconImageUrl Property

C#

```
public string CatalogIconImageUrl;
```

Visual Basic

```
Public Property CatalogIconImageUrl() As string
```

Description

This is CatalogIconImageUrl, a member of class EkWebPart.

EkWebPart.Description Property

C#

```
public string Description;
```

Visual Basic

```
Public Property Description() As string
```

Description

This is Description, a member of class EkWebPart.

EkWebPart.Subtitle Property

C#

```
public string Subtitle;
```

Visual Basic

```
Public Property Subtitle() As string
```

Description

This is Subtitle, a member of class EkWebPart.

EkWebPart.Title Property

C#

```
public string Title;
```

Visual Basic

```
Public Property Title() As string
```

Description

This is Title, a member of class EkWebPart.

EkWebPart.TitleIconImageUrl Property

C#

```
public string TitleIconImageUrl;
```

Visual Basic

```
Public Property TitleIconImageUrl() As string
```

Description

This is TitleIconImageUrl, a member of class EkWebPart.

EkWebPart.TitleUrl Property

C#

```
public string TitleUrl;
```

Visual Basic

```
Public Property TitleUrl() As string
```

Description

This is TitleUrl, a member of class EkWebPart.

EkWebPart.WebBrowsableObject Property

C#

```
[Browsable(false)]
public object WebBrowsableObject;
```

Visual Basic

```
<Browsable(false)>
Public Property WebBrowsableObject() As object
```

Description

This is WebBrowsableObject, a member of class EkWebPart.

EkXsltBase Class

Summary description for EkXsltBase.

Class Hierarchy

C#

```
public abstract class EkXsltBase : EkWebControl;
```

Visual Basic

```
Public abstract Class EkXsltBase
Inherits EkWebControl
```

File

EkXsltBase.cs

EkXsltBase.EkXsltBase Constructor

Constructor for EkXlstBase

C#

```
public EkXsltBase();
```

Visual Basic

```
Public Sub New()
```

EkXsltBase Properties

The properties of the EkXsltBase class are listed here.

Public Properties

	Name	Description
	CustomXml (see page 232)	This property provides a mechanism to inject custom XML into the generated XML before being processed by the XSLT.
	DisplayXslt (see page 233)	The Xslt to use to render the control (optional).
	EnableErrorLogging (see page 233)	Enables or disables logging the error to the event log.
	XmlDoc (see page 233)	XmlDocument used in this control.
	XsltDoc (see page 234)	Xslt used in this control.

EkXsltBase.CustomXml Property

This property provides a mechanism to inject custom XML into the generated XML before being processed by the XSLT.

C#

```
[Browsable(false)]
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("This property provides a mechanism to inject custom XML into the generated XML
```

```
before being processed by the XSLT.")]
[DefaultValue("")]
public string CustomXml;
```

Visual Basic

```
<Browsable(false)>
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("This property provides a mechanism to inject custom XML into the generated XML
before being processed by the XSLT.")>
<DefaultValue("")>
Public Property CustomXml() As string
```

EkXsltBase.DisplayXslt Property

The Xslt to use to render the control (optional).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the Control (optional)")]
[TypeConverter(typeof(XsltTypeConverter))]
[DefaultValue("ecmNavigation")]
[Personalizable()]
public virtual string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the Control (optional)")>
<TypeConverter(typeof(XsltTypeConverter))>
<DefaultValue("ecmNavigation")>
<Personalizable()>
Public virtual Property DisplayXslt() As string
```

EkXsltBase.EnableErrorLogging Property

Enables or disables logging the error to the event log.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Enables or disables logging the error to the event log.")]
[DefaultValue("true")]
[Obsolete("EnableErrorLogging property is obsolete, use LogLevel", false)]
public bool EnableErrorLogging;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Enables or disables logging the error to the event log.")>
<DefaultValue("true")>
<Obsolete("EnableErrorLogging property is obsolete, use LogLevel", false)>
Public Property EnableErrorLogging() As bool
```

EkXsltBase.XmlDoc Property

XmlDocument used in this control.

C#

```
[Browsable(false)]
public virtual XmlDocument XmlDoc;
```

Visual Basic

```
<Browsable(false)>
Public virtual Property XmlDoc() As XmlDocument
```

EkXsltBase.XsltDoc Property

Xslt used in this control.

C#

```
[Browsable(false)]
public virtual XmlDocument XsltDoc;
```

Visual Basic

```
<Browsable(false)>
Public virtual Property XsltDoc() As XmlDocument
```

ExplorerDragDrop Class

Defines the Explorer Drag Drop control, used to upload assets via Desktop drag drop to the CMS.

Class Hierarchy

C#

```
[DefaultProperty("Path")]
[ToolboxData("<{0}:ExplorerView runat=\"server\"></{0}:ExplorerView>")]
[AspNetHostingPermission(SecurityAction.Demand, Level = AspNetHostingPermissionLevel.Minimal)]
[System.ComponentModel.ToolboxItem(false)]
public class ExplorerDragDrop : WebControl;
```

Visual Basic

```
<DefaultProperty("Path")>
<ToolboxData("<{0}:ExplorerView runat=\"server\"></{0}:ExplorerView>")>
<AspNetHostingPermission(SecurityAction.Demand, Level = AspNetHostingPermissionLevel.Minimal)>
<System.ComponentModel.ToolboxItem(false)>
Public Class ExplorerDragDrop
Inherits WebControl
```

File

ExplorerDragDrop.cs

ExplorerDragDrop Properties

The properties of the ExplorerDragDrop class are listed here.

Public Properties

	Name	Description
	AssetID (see page 234)	Ektron CMS Asset (see page 822) ID.
	ContentLanguage (see page 235)	Ektron CMS Content (see page 822) Language.
	FolderID (see page 235)	Ektron CMS folder ID.
	IsImage (see page 235)	The ID of the taxonomy where assets are added.
	OverrideExtension (see page 236)	The ID of the taxonomy where assets are added.
	TaxonomyId (see page 236)	Ektron CMS Taxonomy (see page 876) Id.

ExplorerDragDrop.AssetID Property

Ektron CMS Asset (see page 822) ID.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("-1")]
[Localizable(true)]
public string AssetID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("-1")>
<Localizable(true)>
Public Property AssetID() As string
```

ExplorerDragDrop.ContentLanguage Property

Ektron CMS Content (see page 822) Language.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(1033)]
[Localizable(true)]
public int ContentLanguage;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(1033)>
<Localizable(true)>
Public Property ContentLanguage() As Integer
```

ExplorerDragDrop.FolderID Property

Ektron CMS folder ID.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(1)]
[Localizable(true)]
public long FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(1)>
<Localizable(true)>
Public Property FolderID() As long
```

ExplorerDragDrop.IsImage Property

The ID of the taxonomy where assets are added.

C#

```
[Category("Ektron")]
[DefaultValue("0")]
[Description("Used to specify only images.")]
public int IsImage;
```

Visual Basic

```
<Category("Ektron")>
```

```
<DefaultValue("0")>
<Description("Used to specify only images.")>
Public Property IsImage() As Integer
```

ExplorerDragDrop.OverrideExtension Property

The ID of the taxonomy where assets are added.

C#

```
[Category("Ektron")]
[DefaultValue("0")]
[Description("Extension the uploaded file should be.")]
public string OverrideExtension;
```

Visual Basic

```
<Category("Ektron")>
<DefaultValue("0")>
<Description("Extension the uploaded file should be.")>
Public Property OverrideExtension() As string
```

ExplorerDragDrop.TaxonomyId Property

Ektron CMS Taxonomy (see page 876) Id.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(-1)]
[Localizable(true)]
public long TaxonomyId;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(-1)>
<Localizable(true)>
Public Property TaxonomyId() As long
```

Favorites Class

Summary description for Favorites

Class Hierarchy

C#

```
[DefaultProperty("DefaultUserID"), ToolboxData("<{0}:Favorites runat=server></{0}:Favorites>")]
public class Favorites : EkXsltWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultUserID"), ToolboxData("<{0}:Favorites runat=server></{0}:Favorites>")>
Public Class Favorites
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

File

Favorites.cs

Favorites Methods

The methods of the Favorites class are listed here.

Public Methods

	Name	Description
	Fill (see page 237)	Populates the data for the Favorites (see page 236).
	ToString (see page 237)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 237)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Favorites.Fill Method

Populates the data for the Favorites (see page 236).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Favorites.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Favorites.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

Favorites Properties

The properties of the Favorites class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 238)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

	DefaultUserID (see page 239)	The ID for the community group that the control will use.
	DisplayMode (see page 239)	Select the way this control displays community group information. Choices are: **List** - shows a list of Favorites (see page 236) sorted in alphabetical order. **Directory (see page 196)** - this selection allows users to group their favorites by folders. When this selection is chosen, the following menu items are added to the control: • **Add Folder (see page 683)** - allows a user to add a sub folder. • **Move** - allows a user to move content into a folder. This item only appears when there is a folder.
	DisplayXslt (see page 239)	Select the type of review to display, or enter the path to a custom XSLT. (optional).
	DynamicParameter (see page 240)	Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. Leave blank to always display the list summary.
	FavoritesTitle (see page 240)	The title to show at the top of the control.
	FavoritesUriTitle (see page 240)	The title to show at the top of the control.
	Link (see page 240)	Enter a link to the content Web page. This allows a user to click a link in the Favorites (see page 236) control and be taken to the content. There are three variables used within the link that represent the content's ID, the language and the content's title. • **{0}** - represents the content's ID. • **{1}** - represents the content's language. • **{2}** - represents the content's title. You need to have all three variables in the link. The Web form can be relative or absolute. Below is an example. contentpage.aspx?cid={0}&lang={1}&title={2}
	LinkTarget (see page 241)	Determines the type of window that appears on this Web form when a user clicks an item in the Favorites (see page 236) control. The default is _self.
	MarkupLanguage (see page 241)	Enter name of a markup file that manages the display of this server control.
	MaxResults (see page 241)	Maximum number of results to be returned. If set to zero, there is no limit.
	Stylesheet (see page 242)	Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet.
	XmlDoc (see page 242)	Returns the Xslt object for the collection.

Favorites.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
```

```
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

Favorites.DefaultUserID Property

The ID for the community group that the control will use.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID for the user that the control will use.")]
public long DefaultUserID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID for the user that the control will use.")>
Public Property DefaultUserID() As long
```

Favorites.DisplayMode Property

Select the way this control displays community group information. Choices are:

List - shows a list of Favorites (see page 236) sorted in alphabetical order.

Directory (see page 196) - this selection allows users to group their favorites by folders. When this selection is chosen, the following menu items are added to the control:

- **Add Folder (see page 683)** - allows a user to add a sub folder.
- **Move** - allows a user to move content into a folder. This item only appears when there is a folder.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Directory), Description("This
property enables the user to show the favorites either in the List or Directory display mode")]
public eDisplayMode DisplayMode;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Directory), Description("This
property enables the user to show the favorites either in the List or Directory display mode")>
Public Property DisplayMode() As eDisplayMode
```

Favorites.DisplayXslt Property

Select the type of review to display, or enter the path to a custom XSLT. (optional).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltFavoriteTypeConverter))]
[DefaultValue("Builtin")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltFavoriteTypeConverter))>
```

```
<DefaultValue("Builtin")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

Favorites.DynamicParameter Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. Leave blank to always display the list summary.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute. Leave blank to always use the default ID.")]
[Personalizable()]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute. Leave blank to always use the default ID.")>
<Personalizable()>
Public Property DynamicParameter() As string
```

Favorites.FavoritesTitle Property

The title to show at the top of the control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("Favorites"), Description("The title to show
for the Content at the top of the control.")]
public string FavoritesTitle;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("Favorites"), Description("The title to show
for the Content at the top of the control.")>
Public Property FavoritesTitle() As string
```

Favorites.FavoritesUriTitle Property

The title to show at the top of the control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("Favorites"), Description("The title to show
for the Links at the top of the control.")]
public string FavoritesUriTitle;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("Favorites"), Description("The title to show
for the Links at the top of the control.")>
Public Property FavoritesUriTitle() As string
```

Favorites.Link Property

Enter a link to the content Web page. This allows a user to click a link in the Favorites (see page 236) control and be taken to the content. There are three variables used within the link that represent the content's ID, the language and the content's title.

- **{0}** - represents the content's ID.

- **{1}** - represents the content's language.
- **{2}** - represents the content's title.

You need to have all three variables in the link. The Web form can be relative or absolute. Below is an example.

contentpage.aspx?cid={0}&lang={1}&title={2}

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue("Use QuickLink"), Description("The favorite
link. {0} is the favorite ID, {1} is the favorite language, and {2} is the favorite title.") ]
public string Link;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue("Use QuickLink"), Description("The favorite
link. {0} is the favorite ID, {1} is the favorite language, and {2} is the favorite title.") >
Public Property Link() As string
```

Favorites.LinkTarget Property

Determines the type of window that appears on this Web form when a user clicks an item in the Favorites (see page 236) control. The default is _self.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self),
Description("Determines the type of window that appears on this Web form when a user clicks an
item in the Favorites control. The default is _self.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self),
Description("Determines the type of window that appears on this Web form when a user clicks an
item in the Favorites control. The default is _self.") >
Public Property LinkTarget() As ItemLinkTargets
```

Favorites.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

Favorites.MaxResults Property

Maximum number of results to be returned. If set to zero, there is no limit.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of users to fetch per page. If set to zero, there is no
limit.")]
[Personalizable()]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("The maximum number of users to fetch per page. If set to zero, there is no
limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

Favorites.Stylesheet Property

Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Specify the path to a style sheet for use with the sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Specify the path to a style sheet for use with the sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<Personalizable()>
Public Property Stylesheet() As string
```

Favorites.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

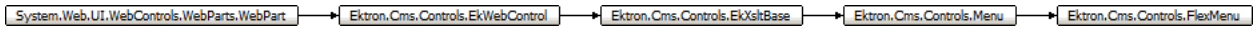
Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

FlexMenu Class

The FlexMenu server control displays a menu on a Web form. Although it looks and acts like a Smart Menu (see page 368), its architecture is substantially different. Specifically, a FlexMenu creates XML. So, you can modify its behavior using an XSLT file, and change its appearance using a cascading style sheet (.css) file.

Class Hierarchy



C#

```
public class FlexMenu : Ektron.Cms.Controls.Menu;
```

Visual Basic

```
Public Class FlexMenu
Inherits Ektron.Cms.Controls.Menu
```

File

FlexMenu.cs

FlexMenu Methods

The methods of the FlexMenu class are listed here.

Public Methods

	Name	Description
	Fill (see page 243)	Calls up the database and updates the contents of the Menu if the data is out of sync with the specified parameters.
	SetDirty (see page 243)	Marks the Control to call the database since output-dependent parameters have been changed.

FlexMenu.Fill Method

Calls up the database and updates the contents of the Menu if the data is out of sync with the specified parameters.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

FlexMenu.SetDirty Method

Marks the Control to call the database since output-dependent parameters have been changed.

C#

```
public new void SetDirty();
```

Visual Basic

```
Public new Function SetDirty() As void
```

FlexMenu Properties

The properties of the FlexMenu class are listed here.

Public Properties

	Name	Description
	AutoCollapseBranches (see page 244)	Set to True to automatically close other menu branches when a new one is opened.
	CacheInterval (see page 244)	Sets the amount of time in seconds that the FlexMenus' data is cached. The default is 600 (10 minutes). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	CacheIntervalAuthenticated (see page 245)	Sets the amount of time in seconds that the FlexMenus' data is cached for an authenticated user. The default is 600 (10 minutes). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheIntervalAuthenticated property to 300 (three hundred).
	DisplayXslt (see page 245)	The Xslt used to render the Menu (see page 368).
	EnableAjax (see page 245)	Set to true to enable Ajax, which only downloads sub-menus as needed.
	EnableMouseOverPopUp (see page 246)	Set to True and the submenus appear as soon as the cursor moves over them.
	EnableSmartOpen (see page 246)	When true, menu will auto-open to show matching IDs or folder associations.
	EnableSubmenuLinkPathMatching (see page 246)	Set to enable auto-submenu opening based on link match (if all else fails).
	EnableSubmenuTemplateMatching (see page 247)	Set to enable auto-submenu opening based on a template filename match (if all else fails).
	LogInfo (see page 247)	Set to true to enable logging verbose info to the event log (such as information on reasons for menu selection), must set web.config LogLevel (see page 228) to verbose.

	MasterControlId (see page 247)	If this control is a slave Flex menu, enter the ID of the master Flex menu control. Otherwise, leave blank.
	MenuDepth (see page 248)	To let site visitors browse through all menu levels, enter zero (0). To restrict site visitors to a menu level, enter the number of the lowest level.
	StartCollapsed (see page 248)	Set to True to initially render menu with submenus closed.
	StartLevel (see page 248)	Enter a number to indicate the level at which you want this menu to display when it first appears. To begin the menu display at the root level, enter zero (0).
	Stylesheet (see page 249)	The Style-Sheet to use for rendering the Menu (see page 368) (optional).
	UseAliasRealId (see page 249)	If system id using manual aliasing, and the aliased menu item does not get selected due to external link that generates a URL which contains the real path and ID, setting this true will force first matching menu item ID to be selected.
	XmlDoc (see page 249)	XmlDocument used in this control.
	XsltDoc (see page 250)	Xslt used in this control.

FlexMenu.AutoCollapseBranches Property

Set to True to automatically close other menu branches when a new one is opened.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to automatically close other menu branches when a new one is
opened.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
public bool AutoCollapseBranches;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to automatically close other menu branches when a new one is
opened.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
Public Property AutoCollapseBranches() As bool
```

FlexMenu.CacheInterval Property

Sets the amount of time in seconds that the FlexMenus' data is cached. The default is 600 (10 minutes). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time in seconds that the FlexMenus' data is cached. The
default is 600 (10 minutes). This is the amount of time, in seconds, a control?s data is
cached. For example, if you want to cache the data for five minutes, you set the CacheInterval
property to 300 (three hundred).")]
[DefaultValue(600.0)]
public override double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time in seconds that the FlexMenus' data is cached. The
default is 600 (10 minutes). This is the amount of time, in seconds, a control?s data is
cached. For example, if you want to cache the data for five minutes, you set the CacheInterval
property to 300 (three hundred).")>
<DefaultValue(600.0)>
Public override Property CacheInterval() As double
```

FlexMenu.CacheIntervalAuthenticated Property

Sets the amount of time in seconds that the FlexMenus' data is cached for an authenticated user. The default is 600 (10 minutes). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheIntervalAuthenticated property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time in seconds that the FlexMenus' data is cached for an
authenticated user. The default is 600 (10 minutes). This is the amount of time, in seconds, a
control?s data is cached. For example, if you want to cache the data for five minutes, you set
the CacheInterval property to 300 (three hundred).")]
[DefaultValue(600.0)]
public double CacheIntervalAuthenticated;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time in seconds that the FlexMenus' data is cached for an
authenticated user. The default is 600 (10 minutes). This is the amount of time, in seconds, a
control?s data is cached. For example, if you want to cache the data for five minutes, you set
the CacheInterval property to 300 (three hundred).")>
<DefaultValue(600.0)>
Public Property CacheIntervalAuthenticated() As double
```

FlexMenu.DisplayXslt Property

The Xslt used to render the Menu (see page 368).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the Menu.")]
[TypeConverter(typeof(StringConverter))]
[DefaultValue("")]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the Menu.")>
<TypeConverter(typeof(StringConverter))>
<DefaultValue("")>
Public override Property DisplayXslt() As string
```

FlexMenu.EnableAjax Property

Set to true to enable Ajax, which only downloads sub-menus as needed.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to true to enable Ajax, which only downloads sub-menus as needed.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool EnableAjax;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to true to enable Ajax, which only downloads sub-menus as needed.")>
```

```
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property EnableAjax() As bool
```

FlexMenu.EnableMouseOverPopUp Property

Set to True and the submenus appear as soon as the cursor moves over them.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True and the submenus appear as soon as the cursor moves over them.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool EnableMouseOverPopUp;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True and the submenus appear as soon as the cursor moves over them.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property EnableMouseOverPopUp() As bool
```

FlexMenu.EnableSmartOpen Property

When true, menu will auto-open to show matching IDs or folder associations.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("When true, menu will auto-open to show matching IDs or folder associations.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
public bool EnableSmartOpen;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("When true, menu will auto-open to show matching IDs or folder associations.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
Public Property EnableSmartOpen() As bool
```

FlexMenu.EnableSubmenuLinkPathMatching Property

Set to enable auto-submenu opening based on link match (if all else fails).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to enable auto-submenu opening based on link match (if all else fails).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool EnableSubmenuLinkPathMatching;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to enable auto-submenu opening based on link match (if all else fails).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
```

```
Public Property EnableSubmenuLinkPathMatching() As bool
```

FlexMenu.EnableSubmenuTemplateMatching Property

Set to enable auto-submenu opening based on a template filename match (if all else fails).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to enable auto-submenu opening based on a template filename match (if all
else fails).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool EnableSubmenuTemplateMatching;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to enable auto-submenu opening based on a template filename match (if all
else fails).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property EnableSubmenuTemplateMatching() As bool
```

FlexMenu.LogInfo Property

Set to true to enable logging verbose info to the event log (such as information on reasons for menu selection), must set web.config LogLevel (see page 228) to verbose.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Used to assist identifying a configuration problem, when the menu does not
appear to select what the designer intended.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool LogInfo;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Used to assist identifying a configuration problem, when the menu does not
appear to select what the designer intended.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property LogInfo() As bool
```

FlexMenu.MasterControlId Property

If this control is a slave Flex menu, enter the ID of the master Flex menu control. Otherwise, leave blank.

C#

```
[Description("If this control is a slave Flex menu, enter the ID of the master Flex menu
control. Otherwise, leave blank.")]
[Bindable(true)]
[TypeConverter(typeof(StringConverter))]
[DefaultValue("")]
public string MasterControlId;
```

Visual Basic

```
<Description("If this control is a slave Flex menu, enter the ID of the master Flex menu
```

```
control. Otherwise, leave blank.")>
<Bindable(true)>
<TypeConverter(typeof(StringConverter))>
<DefaultValue("")>
Public Property MasterControlId() As string
```

FlexMenu.MenuDepth Property

To let site visitors browse through all menu levels, enter zero (0). To restrict site visitors to a menu level, enter the number of the lowest level.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("To let site visitors browse through all menu levels, enter zero (0). To restrict
site visitors to a menu level, enter the number of the lowest level.")]
[TypeConverter(typeof(Int32Converter))]
[DefaultValue(0)]
public int MenuDepth;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("To let site visitors browse through all menu levels, enter zero (0). To restrict
site visitors to a menu level, enter the number of the lowest level.")>
<TypeConverter(typeof(Int32Converter))>
<DefaultValue(0)>
Public Property MenuDepth() As Integer
```

FlexMenu.StartCollapsed Property

Set to True to initially render menu with submenus closed.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to initially render menu with submenus closed.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
public bool StartCollapsed;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to initially render menu with submenus closed.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
Public Property StartCollapsed() As bool
```

FlexMenu.StartLevel Property

Enter a number to indicate the level at which you want this menu to display when it first appears. To begin the menu display at the root level, enter zero (0).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Enter a number to indicate the level at which you want this menu to display when
it first appears. To begin the menu display at the root level, enter zero (0).")]
[TypeConverter(typeof(Int32Converter))]
[DefaultValue(1)]
public int StartLevel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Enter a number to indicate the level at which you want this menu to display when
it first appears. To begin the menu display at the root level, enter zero (0).")>
<TypeConverter(typeof(Int32Converter))>
<DefaultValue(1)>
Public Property StartLevel() As Integer
```

FlexMenu.Stylesheet Property

The Style-Sheet to use for rendering the Menu (see page 368) (optional).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Style-Sheet to use for rendering the Menu.")]
[TypeConverter(typeof(StringConverter))]
[DefaultValue("")]
public override string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Style-Sheet to use for rendering the Menu.")>
<TypeConverter(typeof(StringConverter))>
<DefaultValue("")>
Public override Property Stylesheet() As string
```

FlexMenu.UseAliasRealId Property

If system id using manual aliasing, and the aliased menu item does not get selected due to external link that generates a URL which contains the real path and ID, setting this true will force first matching menu item ID to be selected.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Not normally needed (default: False), but available for special cases when both
alias and non-alias references to the same menu item must be allowed.")]
[DefaultValue("false")]
public bool UseAliasRealId;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Not normally needed (default: False), but available for special cases when both
alias and non-alias references to the same menu item must be allowed.")>
<DefaultValue("false")>
Public Property UseAliasRealId() As bool
```

FlexMenu.XmlDoc Property

XmlDocument used in this control.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

FlexMenu.XsltDoc Property

Xslt used in this control.

C#

```
public override XmlDocument XsltDoc;
```

Visual Basic

```
Public override Property XsltDoc() As XmlDocument
```

FolderBreadcrumb Class

The FolderBreadcrumb server control displays sitemap breadcrumbs. This type of trail is typically made up of the folder path for the content. It is important to note that the FolderBreadcrumb server control does not simply read your folder structure and display its path. It reads breadcrumb information set in the Workarea's folder properties for each folder. Administrators define a folder's breadcrumbs on the Breadcrumb tab located in the Workarea folder properties. *See Also:* The Administrator Manual > "Managing Content (see page 822)" > "Managing Content (see page 822) Folders" > "Breadcrumbs"

For additional information on this server control, see the Developer Manual section "Introduction to Ektron CMS400.NET Server Controls (see page 7)" > "FolderBreadcrumb Server Control"

Class Hierarchy

C#

```
[DefaultProperty("Text"), ToolboxData("<{0}:FolderBreadcrumb
runat=\"server\"></{0}:FolderBreadcrumb>")]
public class FolderBreadcrumb : EkXsltBase;
```

Visual Basic

```
<DefaultProperty("Text"), ToolboxData("<{0}:FolderBreadcrumb
runat=\"server\"></{0}:FolderBreadcrumb>")>
Public Class FolderBreadcrumb
Inherits EkXsltBase
```

File

FolderBreadcrumb.cs

FolderBreadcrumb Methods

The methods of the FolderBreadcrumb class are listed here.

Public Methods

	Name	Description
	Fill (see page 250)	Method that can be overridden and used to grab data from the back end for each server control.

FolderBreadcrumb.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

FolderBreadcrumb Properties

The properties of the FolderBreadcrumb class are listed here.

Public Properties

	Name	Description
	DefaultContentID (see page 251)	Gets or sets a Content (see page 822) ID and displays the breadcrumb trail for the folder in which the content resides. To use this property, breadcrumb information for a folder must be defined in the Workarea > [Specified Folder (see page 683)] > Folder (see page 683) properties > Breadcrumb tab.
	DefaultFolderID (see page 251)	The folder ID for which you want the breadcrumb trail to display. If a DefaultContentID (see page 251) is given then it overrides this property. To use this property, breadcrumb information for a folder must be defined in the Workarea > [Specified Folder (see page 683)] > Folder (see page 683) properties > Breadcrumb tab.
	DisplayStyle (see page 252)	Indicate how to display the breadcrumb trail: horizontally or vertically. The default is Horizontal.
	DisplayXslt (see page 252)	Select the type of review to display, or enter the path to a custom XSLT. (optional).
	DynamicParameter (see page 253)	Gets or Sets the QueryString parameter to read a Content (see page 822) ID dynamically - Set to String.Empty if you wish to always display the default ContentID (static.)
	LinkTarget (see page 253)	Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self. • **_Self** - opens link in the same window. • **_Top** - opens link in a parent window. • **_Blank** - opens link in a new window. • **_Parent** - opens link in the parent frame.
	Mode (see page 253)	Allows you to make the breadcrumb trail appear as non-hyperlinked plain text. The default is Normal. • **Normal** = breadcrumb trail is hyperlinked • **DisplayOnly** = breadcrumb trail is plain text
	separator (see page 254)	Enter one or more characters to separate the items in a breadcrumb trail.
	XmlDoc (see page 254)	Returns the Xslt object for the collection.

FolderBreadcrumb.DefaultContentID Property

Gets or sets a Content (see page 822) ID and displays the breadcrumb trail for the folder in which the content resides. To use this property, breadcrumb information for a folder must be defined in the Workarea > [Specified Folder (see page 683)] > Folder (see page 683) properties > Breadcrumb tab.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or sets the Content ID and displays the breadcrumb trail for the folder in
which the content resides.")]
public long DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or sets the Content ID and displays the breadcrumb trail for the folder in
which the content resides.")>
Public Property DefaultContentID() As long
```

FolderBreadcrumb.DefaultFolderID Property

The folder ID for which you want the breadcrumb trail to display. If a DefaultContentID (see page 251) is given then it overrides this property. To use this property, breadcrumb information for a folder must be defined in the Workarea > [Specified Folder (

see page 683)] > Folder (see page 683) properties > Breadcrumb tab.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The folder ID for which you want the breadcrumb trail to display. If a
DefaultContentID is given then it overrides this property.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long DefaultFolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The folder ID for which you want the breadcrumb trail to display. If a
DefaultContentID is given then it overrides this property.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property DefaultFolderID() As long
```

FolderBreadcrumb.DisplayStyle Property

Indicate how to display the breadcrumb trail: horizontally or vertically. The default is Horizontal.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(DisplayStyles.Horizontal),
Description("Indicate how to display the breadcrumb trail: horizontally or vertically. The
default is Horizontal."), ]
public new DisplayStyles DisplayStyle;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(DisplayStyles.Horizontal),
Description("Indicate how to display the breadcrumb trail: horizontally or vertically. The
default is Horizontal."), >
Public new Property DisplayStyle() As DisplayStyles
```

FolderBreadcrumb.DisplayXslt Property

Select the type of review to display, or enter the path to a custom XSLT. (optional).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltFolderBreadTypeConverter))]
[DefaultValue("Default")]
[Personalizable()]
[Browsable(true)]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltFolderBreadTypeConverter))>
<DefaultValue("Default")>
<Personalizable()>
<Browsable(true)>
Public override Property DisplayXslt() As string
```

FolderBreadcrumb.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 822) ID dynamically - Set to String.Empty if you wish to always display the default ContentID (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Content ID dynamically - Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Content ID dynamically - Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

FolderBreadcrumb.LinkTarget Property

Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self.

- **_Self** - opens link in the same window.
- **_Top** - opens link in a parent window.
- **_Blank** - opens link in a new window.
- **_Parent** - opens link in the parent frame.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

FolderBreadcrumb.Mode Property

Allows you to make the breadcrumb trail appear as non-hyperlinked plain text. The default is Normal.

- **Normal** = breadcrumb trail is hyperlinked
- **DisplayOnly** = breadcrumb trail is plain text

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(Modes.Normal), Description("Set to
\"display only\" mode to render items in the Trail without adding links to it.") ]
public Modes Mode;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(Modes.Normal), Description("Set to
\"display only\" mode to render items in the Trail without adding links to it.") >
Public Property Mode() As Modes
```

FolderBreadcrumb.separator Property

Enter one or more characters to separate the items in a breadcrumb trail.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(" > "), Description("Enter one or more
characters to separate the items in a breadcrumb trail.") ]
public string separator;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(" > "), Description("Enter one or more
characters to separate the items in a breadcrumb trail.") >
Public Property separator() As string
```

FolderBreadcrumb.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

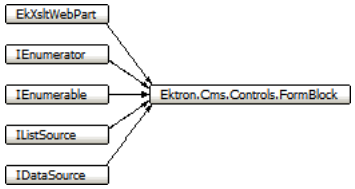
Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

FormBlock Class

The FormBlock server control displays a content block associated with a form.

Class Hierarchy



C#

```
[DefaultProperty("DefaultFormID"), ToolboxData("<{0}:FormBlock runat=server></{0}:FormBlock>")]
public class FormBlock : EkXsltWebPart, IEnumerator, IEnumerable, IListSource, IDataSource;
```

Visual Basic

```
<DefaultProperty("DefaultFormID"), ToolboxData("<{0}:FormBlock runat=server></{0}:FormBlock>")>
Public Class FormBlock
Inherits EkXsltWebPart
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
```

File

FormBlock.cs

FormBlock Methods

The methods of the FormBlock class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 255)	Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.
	Fill (see page 255)	Overrideable method that can be used to grab data from the back end for each server control.
	GetEnumerator (see page 255)	Gets a System.Collections.IEnumerator for the collection.
	GetView (see page 255)	Gets the view data using a dataset.
	GetViewNames (see page 256)	Gets the view names.
	MoveNext (see page 256)	Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.
	Reset (see page 256)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 256)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.
	ToString (see page 256)	Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.
	Wrap (see page 257)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

FormBlock.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

C#

```
public void CopyTo(Array array, int i);
```

Visual Basic

```
Public Function CopyTo(array As Array, i As Integer) As void
```

FormBlock.Fill Method

Overrideable method that can be used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

FormBlock.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

```
public IEnumerator GetEnumerator();
```

Visual Basic

```
Public Function GetEnumerator() As IEnumerator
```

FormBlock.GetView Method

Gets the view data using a dataset.

C#

```
public DataSourceView GetView(string viewName);
```

Visual Basic

```
Public Function GetView(viewName As string) As DataSourceView
```

Parameters

Parameters	Description
string viewName	

Returns

DataSourceView

FormBlock.GetViewNames Method

Gets the view names.

C#

```
public ICollection GetViewNames();
```

Visual Basic

```
Public Function GetViewNames() As ICollection
```

Returns

View names

FormBlock.MoveNext Method

Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.

C#

```
public bool MoveNext();
```

Visual Basic

```
Public Function MoveNext() As bool
```

Returns

Boolean indicating if the move was within the range of the array.

FormBlock.Reset Method

Brings the index of the collection back to zero. Does not clear the collection.

C#

```
public void Reset();
```

Visual Basic

```
Public Function Reset() As void
```

FormBlock.this Indexer

Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.

C#

```
public Ektron.Cms.Controls.ContentItem.ContentRow this[int i];
```

Visual Basic

```
Public Sub this(i As Integer)
```

FormBlock.ToString Method

Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

FormBlock.Wrap Method

Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

FormBlock Properties

The properties of the FormBlock class are listed here.

Public Properties

	Name	Description
	AddValidation (see page 257)	Set this value to True to add the validation to your main runat=server form.
	CacheInterval (see page 258)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	Count (see page 258)	Indicates the number of elements in your Collection (see page 95).
	Current (see page 258)	Gets the current object in the collection.
	DefaultFormID (see page 259)	Gets or Sets the Default FormID for this Control.
	DynamicParameter (see page 259)	Gets or sets the QueryString parameter to read a Form (see page 871) ID dynamically - Set to String.Empty if you wish to always display the default FormID (static.)
	EkItem (see page 259)	Gets the Ektron.Cms.Common.ContentBase for the current Form (see page 871) Block.
	Fields (see page 260)	Returns a collection of fields defined in the selected form.
	FormName (see page 260)	Set this value to the name of your main runat=server form.
	IncludeTags (see page 260)	Set to True to have this Ektron Form (see page 871) element to wrap its own Form (see page 871) tags around itself.
	IsSynchronized (see page 261)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	StartingFolderID (see page 261)	WEB PARTS ONLY: Allows a developer to set the starting folder for the personalization/collaboration selection dialog.
	SyncRoot (see page 261)	Gets an object that can be used to synchronize the collection.

FormBlock.AddValidation Property

Set this value to True to add the validation to your main runat=server form.

C#

```
[Bindable(true)]
```

```
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set this value to True to add the validation to your main runat=server form.")]
[Personalizable()]
[Obsolete("Please remove form validation using the editor.")]
public bool AddValidation;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set this value to True to add the validation to your main runat=server form.")>
<Personalizable()>
<Obsolete("Please remove form validation using the editor.")>
Public Property AddValidation() As bool
```

FormBlock.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

FormBlock.Count Property

Indicates the number of elements in your Collection (see page 95).

C#

```
[Browsable(false)]
public int Count;
```

Visual Basic

```
<Browsable(false)>
Public Property Count() As Integer
```

FormBlock.Current Property

Gets the current object in the collection.

C#

```
[Browsable(false)]
public object Current;
```

Visual Basic

```
<Browsable(false)>
Public Property Current() As object
```

FormBlock.DefaultFormID Property

Gets or Sets the Default FormID for this Control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[ParenthesizePropertyName(true)]
[Editor(typeof(FormUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("The default ID of the Form Block to display.")]
[Personalizable()]
public long DefaultFormID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<ParenthesizePropertyName(true)>
<Editor(typeof(FormUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("The default ID of the Form Block to display.")>
<Personalizable()>
Public Property DefaultFormID() As long
```

FormBlock.DynamicParameter Property

Gets or sets the QueryString parameter to read a Form (see page 871) ID dynamically - Set to String.Empty if you wish to always display the default FormID (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[TypeConverter(typeof(DynamicFormIDTypeConverter))]
[Description("Gets or sets the QueryString parameter to read a Form ID dynamically - Set to
String.Empty if you wish to always display the default FormID (static.)")]
[Personalizable()]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("None - Use Default")>
<TypeConverter(typeof(DynamicFormIDTypeConverter))>
<Description("Gets or sets the QueryString parameter to read a Form ID dynamically - Set to
String.Empty if you wish to always display the default FormID (static.)")>
<Personalizable()>
Public Property DynamicParameter() As string
```

FormBlock.EkItem Property

Gets the Ektron.Cms.Common.ContentBase for the current Form (see page 871) Block.

C#

```
[Browsable(false)]
public ContentBase EkItem;
```

Visual Basic

```
<Browsable(false)>
Public Property EkItem() As ContentBase
```

FormBlock.Fields Property

Returns a collection of fields defined in the selected form.

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("List of fields defined for this form.")]
public FormFieldCollection Fields;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("List of fields defined for this form.")>
Public Property Fields() As FormFieldCollection
```

FormBlock.FormName Property

Set this value to the name of your main runat=server form.

C#

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Set this value to the name of your main runat=server form.")]
[Personalizable()]
public string FormName;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Set this value to the name of your main runat=server form.")>
<Personalizable()>
Public Property FormName() As string
```

FormBlock.IncludeTags Property

Set to True to have this Ektron Form (see page 871) element to wrap its own Form (see page 871) tags around itself.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to true to have this Ektron Form element to wrap its own Form tags around
itself.")]
[Personalizable()]
public bool IncludeTags;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to have this Ektron Form element to wrap its own Form tags around
```

```
itself.")>
<Personalizable()>
Public Property IncludeTags() As bool
```

FormBlock.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

```
[Browsable(false)]
public bool IsSynchronized;
```

Visual Basic

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

FormBlock.StartingFolderID Property

WEB PARTS ONLY: Allows a developer to set the starting folder for the personalization/collaboration selection dialog.

C#

```
[Bindable(true)]
[Category("Ektron WebParts")]
[Description("WEB PARTS ONLY: Allows a developer to set the starting folder for the
personalization/collaboration selection dialog.")]
[DefaultValue(0)]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable()]
public long StartingFolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron WebParts")>
<Description("WEB PARTS ONLY: Allows a developer to set the starting folder for the
personalization/collaboration selection dialog.")>
<DefaultValue(0)>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable()>
Public Property StartingFolderID() As long
```

FormBlock.SyncRoot Property

Gets an object that can be used to synchronize the collection.

C#

```
[Browsable(false)]
public object SyncRoot;
```

Visual Basic

```
<Browsable(false)>
Public Property SyncRoot() As object
```

FormFieldItem Class

Defines a field within the form.

Class Hierarchy

Ektron.Cms.Controls.FormFieldItem

C#

```
[DefaultProperty("FieldName")]
[ValidationProperty("Value")]
```

```
[TypeConverter(typeof(FormFieldItemConverter))]
public class FormFieldItem;
```

Visual Basic

```
<DefaultProperty("FieldName")>
<ValidationProperty("Value")>
<TypeConverter(typeof(FormFieldItemConverter))>
Public Class FormFieldItem
```

File

FormField.cs

FormFieldItem Properties

The properties of the FormFieldItem class are listed here.

Public Properties

	Name	Description
	Basetype (see page 262)	Gets the basic datatype of the field.
	ContentType (see page 262)	Gets the type of content allowed within the field. This is loosely associated with the datatype.
	DatalistItems (see page 263)	Gets the list of options for this field's value, if applicable, otherwise null/Nothing.
	Datatype (see page 263)	Gets the datatype of the field.
	DisplayName (see page 263)	Gets the name of the field to be displayed.
	FieldName (see page 263)	Gets the name of the field.
	Language (see page 264)	Gets the language code of the datalist. For example, en or en-US.
	Value (see page 264)	Gets or set the value of this field.

FormFieldItem.Basetype Property

Gets the basic datatype of the field.

C#

```
[Category("Data")]
[Description("Basic type of the field's data value, for example, 'text'.")]
public string Basetype;
```

Visual Basic

```
<Category("Data")>
<Description("Basic type of the field's data value, for example, 'text'.")>
Public Property Basetype() As string
```

Remarks

The possible values of the basic datatype vary, including, text, textbox, calendar, and checkbox.

FormFieldItem.ContentType Property

Gets the type of content allowed within the field. This is loosely associated with the datatype.

C#

```
[Category("Data")]
[Description("Type of content allowed within the field")]
public string ContentType;
```

Visual Basic

```
<Category("Data")>
<Description("Type of content allowed within the field")>
Public Property ContentType() As string
```

Remarks

The possible values for content vary.

FormFieldItem.DatalistItems Property

Gets the list of options for this field's value, if applicable, otherwise null/Nothing.

C#

```
[Browsable(false)]
[Category("Data")]
[Description("List of options for this field's value, if applicable, otherwise null/Nothing.")]
public List<FormFieldItemDatalistItem> DatalistItems;
```

Visual Basic

```
<Browsable(false)>
<Category("Data")>
<Description("List of options for this field's value, if applicable, otherwise null/Nothing.")>
Public Property DatalistItems() As List<FormFieldItemDatalistItem>
```

FormFieldItem.Datatype Property

Gets the datatype of the field.

C#

```
[Category("Data")]
[Description("Type of the field's data value, generally conforming to W3C XML Schema types.")]
public string Datatype;
```

Visual Basic

```
<Category("Data")>
<Description("Type of the field's data value, generally conforming to W3C XML Schema types.")>
Public Property Datatype() As string
```

Remarks

The possible values of the datatype vary. Generally they conform to W3C XML Schema types such as 'string' and 'date'. Other values include 'selection' and 'choice'.

FormFieldItem.DisplayName Property

Gets the name of the field to be displayed.

C#

```
[Category("Appearance")]
[Description("Name of the field suitable for display.")]
public string DisplayName;
```

Visual Basic

```
<Category("Appearance")>
<Description("Name of the field suitable for display.")>
Public Property DisplayName() As string
```

Remarks

The name is suitable for human reading.

FormFieldItem.FieldName Property

Gets the name of the field.

C#

```
[Category("Data")]
[Description("Name of the field suitable for XML or programmatic processing.")]
public string FieldName;
```

Visual Basic

```
<Category("Data")>
<Description("Name of the field suitable for XML or programmatic processing.")>
Public Property FieldName() As string
```

Remarks

The name is suitable for machine processing, such as with an XML parser.

FormFieldItem.Language Property

Gets the language code of the datalist. For example, en or en-US.

C#

```
[Category("Data")]
[Description("Language code of the datalist. For example, en or en-US.")]
public string Language;
```

Visual Basic

```
<Category("Data")>
<Description("Language code of the datalist. For example, en or en-US.")>
Public Property Language() As string
```

Remarks

The first two letters of the language code conform to ISO 639. The last two letters are an optional country code that conforms to ISO 3166.

FormFieldItem.Value Property

Gets or set the value of this field.

C#

```
[Browsable(false)]
[Category("Data")]
[Description("The value of this field.")]
public string Value;
```

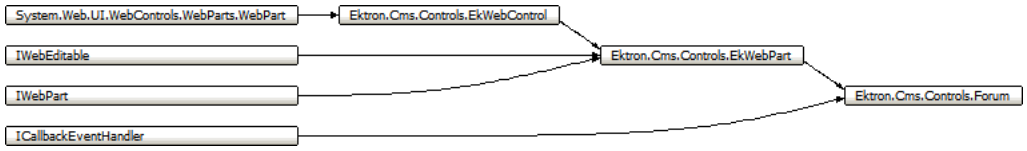
Visual Basic

```
<Browsable(false)>
<Category("Data")>
<Description("The value of this field.")>
Public Property Value() As string
```

Forum Class

The Forum server control displays a discussion board on a Web page.

Class Hierarchy



C#

```
[DefaultProperty("BoardID"), ToolboxData("<{0}:Forum runat=server></{0}:Forum>")]
public class Forum : EkWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("BoardID"), ToolboxData("<{0}:Forum runat=server></{0}:Forum>")>
Public Class Forum
Inherits EkWebPart
Inherits ICallbackEventHandler
```

File

Forum.cs

Forum Methods

The methods of the Forum class are listed here.

Public Methods

	Name	Description
	Fill (see page 265)	Method that can be overridden and used to grab data from the back end for each server control.
	GetCallbackResult (see page 265)	Returns search results to the client callback function
	RaiseCallbackEvent (see page 265)	Processes the callback event by performing the search.
	ToString (see page 266)	Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.
	Wrap (see page 266)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Forum.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Forum.GetCallbackResult Method

Returns search results to the client callback function

C#

```
public string GetCallbackResult();
```

Visual Basic

```
Public Function GetCallbackResult() As string
```

Returns

String containing search results.

Forum.RaiseCallbackEvent Method

Processes the callback event by performing the search.

C#

```
public void RaiseCallbackEvent(string eventArgs);
```

Visual Basic

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

Parameters

Parameters	Description
string eventArgs	Name-value pairs representing form fields passed by client callback function

Forum.ToString Method

Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Forum.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

Forum Properties

The properties of the Forum class are listed here.

Public Properties

	Name	Description
	AllowAnonymousPost (see page 268)	If set to false, will not allow anonymous post on the forum. Defaults to false.
	BoardID (see page 268)	The ID of the Discussion Board to display on this page if one is not defined in a query string parameter.
V	CacheInterval (see page 269)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	CustomOrderBy (see page 269)	Order search results by a custom property. Set this property to the friendly name of the property defined in the Indexing Service. If an invalid property is specified, no search results are returned. If this is left empty, the OrderBy (see page 276) property is used to order search results.
	DefaultObjectID (see page 269)	The object ID for the object that the board will be drawn for.
	DynamicForumParameter (see page 270)	Gets or Sets the QueryString parameter to read a Forum (see page 264) ID dynamically.
	DynamicGroupParameter (see page 270)	Gets or Sets the QueryString parameter to read a Forum (see page 264) ID dynamically.
	DynamicObjectParameter (see page 270)	Checks the QueryString for this value and uses that value when specified. Leave blank to always display the default object id.

	DynamicParameter (see page 271)	Gets or Sets the QueryString parameter to read a Blog (see page 32) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 32) (static.)
	DynamicThreadParameter (see page 271)	Gets or Sets the QueryString parameter to read a Forum (see page 264) ID dynamically.
	EditorCSS (see page 271)	Set the stylesheet on the editor.
	EditorToolbar (see page 271)	Decide which buttons are available in the Editor. For additional information on the Toolbar Editor, See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Adding and Removing Toolbar Buttons From the Editor
	EnableForumQuickSelect (see page 272)	If set to true, will display a drop down list of all forums in the current board. Defaults to false.
	EnableThreadRating (see page 272)	If set to true, will display a 5 star rating system for users to rate a specific thread in a forum.
	ExcludeList (see page 273)	Enter a comma-separated list of custom user properties to exclude from the Forum (see page 264)'s profile page. For example, to suppress the Subscriptions field value, enter **subscriptions**.
	FilterXslt (see page 273)	Create your own custom xslt to filter forum content like HTML Attributes,Tags (see page 805) and unwanted words. See the CMS400.NET Developer Manual section
	HideUserProfile (see page 273)	Set to True to suppress user profiles. The default is True. • **True** - hides user profiles from non-administrator users • **False** - displays user profiles from non-administrator users. However, an administrator can view a profile regardless of how this property is set. On the other hand, if the Private Profile checkbox is checked for any user, his profile information is not visible, regardless of this setting. When the profile is visible, only properties and their values not listed in the `ExcludeList (see page 273)` property appear.
	HighlightedUserRole (see page 274)	Set to True to suppress user profiles. The default is True. • **True** - hides user profiles from non-administrator users • **False** - displays user profiles from non-administrator users. However, an administrator can view a profile regardless of how this property is set. On the other hand, if the Private Profile checkbox is checked for any user, his profile information is not visible, regardless of this setting. When the profile is visible, only properties and their values not listed in the `ExcludeList (see page 273)` property appear.
	JavascriptEditorHeight (see page 274)	Set the height in pixels for the editor. The default is 400.
	JavascriptEditorHTMLMode (see page 275)	Set to True to allow users to edit their posts and replies in HTML. The default is False. See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Allowing HTML Editing in the Javascript Editor"
	JavascriptEditorToolbar (see page 275)	JavaScriptEditor has been deprecated. Instead use EditorToolbar (see page 271) property to Add/Remove buttons in CMS400WebEditor."
	JavascriptEditorWidth (see page 276)	Set the width in pixels for the editor. The default is 625.
	ObjectType (see page 276)	Specifies the type of object to retrieve a board for. The default is DiscussionBoard.
	OrderBy (see page 276)	The order of forum search results. Choices are: • **Title (see page 231)** - The content title (alphabetical) • **ID** - The content ID number • **Date Created** - The date the content was created • **Date Modified** - The date the content was most recently modified • **Editor** - The user who last edited the content (alphabetical) • **Rank** - The rank assigned to the content. See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 725)" > "Display of Search (see page 474) Results" > "Search (see page 474) Result Ranking"

	OrderDirection (see page 277)	Specify the order direction of forum search results. Choices are **Ascending** or **Descending**.
	ProfileLink (see page 277)	Enter a link to the user profile page. This allows a user to click a link in the control and be taken to the user's profile page. There is a variables used within the link that represent the user's ID. **{0}** - represents the user's ID. The Web form can be relative or absolute. Below is an example. `userprofilepage.aspx?uid={0}`
	ProfileLinkTarget (see page 278)	Gets or Sets the target attribute for the user profile links.
	ShowCategories (see page 278)	If set to **True**, taxonomy categories will also be searched for the search text specified. The default is False. **True** = Search (see page 474) across taxonomy categories **False** = Do not search across taxonomy categories
	TaxCategories (see page 278)	Is server control in workarea
	TaxonomyCols (see page 278)	Enter the number of columns in which this taxonomy/category will appear on the page.
	TaxonomyMarkupLanguage (see page 279)	Enter name of a markup file that manages the display of the directory server control.
	Theme (see page 279)	Enter the custom theme's folder name. The theme can be relative to the site root or located in the current folder. If you do not specify a theme, the property uses the location defined by the Discussion Board's CSS theme property in the Workarea.

Forum.AllowAnonymousPost Property

If set to false, will not allow anonymous post on the forum. Defaults to false.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("If set to false, will not allow anonymous post on the forum. Defaults to true.")]
[DefaultValue(true)]
[Personalizable(), WebBrowsable()]
public bool AllowAnonymousPost;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("If set to false, will not allow anonymous post on the forum. Defaults to true.")>
<DefaultValue(true)>
<Personalizable(), WebBrowsable()>
Public Property AllowAnonymousPost() As bool
```

Forum.BoardID Property

The ID of the Discussion Board to display on this page if one is not defined in a query string parameter.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the Discussion Board to display on this page if one is not defined in
a query string parameter.")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long BoardID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the Discussion Board to display on this page if one is not defined in
a query string parameter.")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
```

```
<Personalizable(), WebBrowsable()>
Public Property BoardID() As long
```

Forum.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable(), WebBrowsable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable(), WebBrowsable()>
Public virtual Property CacheInterval() As double
```

Forum.CustomOrderBy Property

Order search results by a custom property. Set this property to the friendly name of the property defined in the Indexing Service. If an invalid property is specified, no search results are returned. If this is left empty, the OrderBy (see page 276) property is used to order search results.

C#

```
[Category("Ektron"), DefaultValue(""), Description("Order search results by a custom property.
Set this property to the friendly name of the property.")]
public string CustomOrderBy;
```

Visual Basic

```
<Category("Ektron"), DefaultValue(""), Description("Order search results by a custom property.
Set this property to the friendly name of the property.")>
Public Property CustomOrderBy() As string
```

Forum.DefaultObjectID Property

The object ID for the object that the board will be drawn for.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID for the object that the board will be drawn for.")]
public long DefaultObjectID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
```

```
<Description("The ID for the object that the board will be drawn for.")>
Public Property DefaultObjectID() As long
```

Forum.DynamicForumParameter Property

Gets or Sets the QueryString parameter to read a Forum (see page 264) ID dynamically.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Forum ID dynamically.")]
[DefaultValue("f")]
public string DynamicForumParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Forum ID dynamically.")>
<DefaultValue("f")>
Public Property DynamicForumParameter() As string
```

Forum.DynamicGroupParameter Property

Gets or Sets the QueryString parameter to read a Forum (see page 264) ID dynamically.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Group ID dynamically.")]
[DefaultValue("g")]
public string DynamicGroupParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Group ID dynamically.")>
<DefaultValue("g")>
Public Property DynamicGroupParameter() As string
```

Forum.DynamicObjectParameter Property

Checks the QueryString for this value and uses that value when specified. Leave blank to always display the default object id.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and uses that value when specified. Leave
blank to always use the default object ID.")]
[Personalizable()]
public string DynamicObjectParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and uses that value when specified. Leave
blank to always use the default object ID.")>
<Personalizable()>
Public Property DynamicObjectParameter() As string
```

Forum.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Blog (see page 32) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 32) (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Forum ID dynamically. Set to
\"Use Default\" if you wish to always display the default Forum (static.)")]
[TypeConverter(typeof(DynamicBoardIDTypeConverter))]
[DefaultValue("boardid")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Forum ID dynamically. Set to
\"Use Default\" if you wish to always display the default Forum (static.)")>
<TypeConverter(typeof(DynamicBoardIDTypeConverter))>
<DefaultValue("boardid")>
Public Property DynamicParameter() As string
```

Forum.DynamicThreadParameter Property

Gets or Sets the QueryString parameter to read a Forum (see page 264) ID dynamically.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Thread ID dynamically.")]
[DefaultValue("t")]
public string DynamicThreadParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Thread ID dynamically.")>
<DefaultValue("t")>
Public Property DynamicThreadParameter() As string
```

Forum.EditorCSS Property

Set the stylesheet on the editor.

C#

```
[Bindable(false)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Set stylesheet to the editor")]
public string EditorCSS;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Set stylesheet to the editor")>
Public Property EditorCSS() As string
```

Forum.EditorToolbar Property

Decide which buttons are available in the Editor. For additional information on the Toolbar Editor, See the CMS400.NET

Developer Manual section "Discussion Board Server Controls (see page 7)" > "Adding and Removing Toolbar Buttons From the Editor

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("StyleMenu, FontMenu, ParagraphMenu, TextFormatMenu, LinkMenu, ClipBoardMenu,
SymbolsMenu, EmoticonSelect, WMV, Table")]
[Description("Decide which buttons are available in the Forum Editor. StyleMenu|Styles;
FontMenu|FontSize,FontName,FontColor;ParagraphMenu|JustifyLeft,JustifyCenter,JustifyRight,Justi
fyFull,BulletedList,NumberedList,Indent,Outdent;
TextFormatMenu|Bold,Italic,Underline,Strikethrough;Superscript,Subscript;
LinkMenu|CreateLink,Unlink;ClipBoardMenu|Cut,Copy,Paste;Undo,Redo,Print; SymbolsMenu;
EmoticonSelect; WMV; Table")]
[Personalizable(), WebBrowsable()]
public string EditorToolbar;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("StyleMenu, FontMenu, ParagraphMenu, TextFormatMenu, LinkMenu, ClipBoardMenu,
SymbolsMenu, EmoticonSelect, WMV, Table")>
<Description("Decide which buttons are available in the Forum Editor. StyleMenu|Styles;
FontMenu|FontSize,FontName,FontColor;ParagraphMenu|JustifyLeft,JustifyCenter,JustifyRight,Justi
fyFull,BulletedList,NumberedList,Indent,Outdent;
TextFormatMenu|Bold,Italic,Underline,Strikethrough;Superscript,Subscript;
LinkMenu|CreateLink,Unlink;ClipBoardMenu|Cut,Copy,Paste;Undo,Redo,Print; SymbolsMenu;
EmoticonSelect; WMV; Table")>
<Personalizable(), WebBrowsable()>
Public Property EditorToolbar() As string
```

Forum.EnableForumQuickSelect Property

If set to true, will display a drop down list of all forums in the current board. Defaults to false.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("If set to true, will display a drop down list of all forums in the current
board. Defaults to false.")]
[DefaultValue(false)]
[Personalizable(), WebBrowsable()]
public bool EnableForumQuickSelect;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("If set to true, will display a drop down list of all forums in the current
board. Defaults to false.")>
<DefaultValue(false)>
<Personalizable(), WebBrowsable()>
Public Property EnableForumQuickSelect() As bool
```

Forum.EnableThreadRating Property

If set to true, will display a 5 star rating system for users to rate a specific thread in a forum.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("If set to true, will display a 5 star rating system for users to rate a specific
thread in a forum.")]
[DefaultValue(false)]
[Personalizable(), WebBrowsable()]
```

```
public bool EnableThreadRating;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("If set to true, will display a 5 star rating system for users to rate a specific
thread in a forum.")>
<DefaultValue(false)>
<Personalizable(), WebBrowsable()>
Public Property EnableThreadRating() As bool
```

Forum.ExcludeList Property

Enter a comma-separated list of custom user properties to exclude from the Forum (see page 264)'s profile page. For example, to suppress the Subscriptions field value, enter **subscriptions**.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Enter a comma-separated list of custom user properties to exclude from the
Forum?s profile page. For example, to suppress the Subscriptions field value, enter
\"subscriptions\".")]
[DefaultValue("")]
public string ExcludeList;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Enter a comma-separated list of custom user properties to exclude from the
Forum?s profile page. For example, to suppress the Subscriptions field value, enter
\"subscriptions\".")>
<DefaultValue("")>
Public Property ExcludeList() As string
```

Forum.FilterXslt Property

Create your own custom xslt to filter forum content like HTML Attributes,Tags (see page 805) and unwanted words. See the CMS400.NET Developer Manual section

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("/workarea/xslt/ForumFilter.xslt")]
[Description("")]
[Personalizable(), WebBrowsable()]
public string FilterXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("/workarea/xslt/ForumFilter.xslt")>
<Description("")>
<Personalizable(), WebBrowsable()>
Public Property FilterXslt() As string
```

Parameters

Parameters	Description
Path to Custom Xslt	Specify the path to the xslt file.It can be relative

Forum.HideUserProfile Property

Set to True to suppress user profiles. The default is True.

- **True** - hides user profiles from non-administrator users
- **False** - displays user profiles from non-administrator users. However, an administrator can view a profile regardless of how this property is set.

On the other hand, if the Private Profile checkbox is checked for any user, his profile information is not visible, regardless of this setting.

When the profile is visible, only properties and their values not listed in the `ExcludeList` (see page 273) property appear.

C#

```
[Bindable(false)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Hide users' profile")]
public bool HideUserProfile;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Hide users' profile")>
Public Property HideUserProfile() As bool
```

Forum.HighlightedUserRole Property

Set to True to suppress user profiles. The default is True.

- **True** - hides user profiles from non-administrator users
- **False** - displays user profiles from non-administrator users. However, an administrator can view a profile regardless of how this property is set.

On the other hand, if the Private Profile checkbox is checked for any user, his profile information is not visible, regardless of this setting.

When the profile is visible, only properties and their values not listed in the `ExcludeList` (see page 273) property appear.

C#

```
[Bindable(false)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Set which user roles are highlighted.")]
public int HighlightedUserRole;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Set which user roles are highlighted.")>
Public Property HighlightedUserRole() As Integer
```

Forum.JavascriptEditorHeight Property

Set the height in pixels for the editor. The default is 400.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(400)]
[Description("The pixel height for the Javascript Editor.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
[Personalizable(), WebBrowsable()]
public int JavascriptEditorHeight;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(400)>
<Description("The pixel height for the Javascript Editor.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
<Personalizable(), WebBrowsable()>
Public Property JavascriptEditorHeight() As Integer
```

Forum.JavascriptEditorHTMLMode Property

Set to True to allow users to edit their posts and replies in HTML. The default is False. See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Allowing HTML Editing in the Javascript Editor"

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to True to allow users to edit their posts and replies in HTML. The default
is False. See the CMS400.NET Developer Manual section \"Discussion Board Server Controls\" >
\"Allowing HTML Editing in the Javascript Editor\"")]
[Obsolete("JavaScript Editor has been deprecated. Instead control is using CMS400WebEditor.")]
[Personalizable(), WebBrowsable()]
public bool JavascriptEditorHTMLMode;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to True to allow users to edit their posts and replies in HTML. The default
is False. See the CMS400.NET Developer Manual section \"Discussion Board Server Controls\" >
\"Allowing HTML Editing in the Javascript Editor\"")>
<Obsolete("JavaScript Editor has been deprecated. Instead control is using CMS400WebEditor.")>
<Personalizable(), WebBrowsable()>
Public Property JavascriptEditorHTMLMode() As bool
```

Forum.JavascriptEditorToolbar Property

JavaScriptEditor has been deprecated. Instead use EditorToolbar (see page 271) property to Add/Remove buttons in CMS400WebEditor."

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("ParagraphMenu,FontFacesMenu,FontSizesMenu,FontForeColorsMenu|Bold,Italic,Underli
ne,Strikethrough;Superscript,Subscript,RemoveFormat|JustifyLeft,JustifyCenter,JustifyRight,Just
ifyFull;BulletedList,NumberedList,Indent,Outdent;CreateLink,Unlink,InsertRule|Cut,Copy,Paste;Un
do,Redo,Print")]
[Description("JavaScriptEditor has been deprecated. Instead use EditorToolbar property to
Add/Remove buttons in CMS400WebEditor.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead use EditorToolbar property to
Add/Remove buttons in CMS400WebEditor.")]
[Personalizable(), WebBrowsable()]
public string JavascriptEditorToolbar;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("ParagraphMenu,FontFacesMenu,FontSizesMenu,FontForeColorsMenu|Bold,Italic,Underli
ne,Strikethrough;Superscript,Subscript,RemoveFormat|JustifyLeft,JustifyCenter,JustifyRight,Just
ifyFull;BulletedList,NumberedList,Indent,Outdent;CreateLink,Unlink,InsertRule|Cut,Copy,Paste;Un
do,Redo,Print")>
<Description("JavaScriptEditor has been deprecated. Instead use EditorToolbar property to
```

```
Add/Remove buttons in CMS400WebEditor.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead use EditorToolbar property to
Add/Remove buttons in CMS400WebEditor.")>
<Personalizable(), WebBrowsable()>
Public Property JavascriptEditorToolbar() As string
```

Forum.JavascriptEditorWidth Property

Set the width in pixels for the editor. The default is 625.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(625)]
[Description("The pixel width for the Javascript Editor.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
[Personalizable(), WebBrowsable()]
public int JavascriptEditorWidth;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(625)>
<Description("The pixel width for the Javascript Editor.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
<Personalizable(), WebBrowsable()>
Public Property JavascriptEditorWidth() As Integer
```

Forum.ObjectType Property

Specifies the type of object to retrieve a board for. The default is DiscussionBoard.

C#

```
[Bindable(true)]
[DefaultValue(Ektron.Cms.Common.EkEnumeration.CMSForumTypes.DiscussionBoard)]
[Description("The type of the object that the board will be drawn for.")]
[Category("Ektron")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Order"), WebDescription("Order")]
public Ektron.Cms.Common.EkEnumeration.CMSForumTypes ObjectType;
```

Visual Basic

```
<Bindable(true)>
<DefaultValue(Ektron.Cms.Common.EkEnumeration.CMSForumTypes.DiscussionBoard)>
<Description("The type of the object that the board will be drawn for.")>
<Category("Ektron")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Order"), WebDescription("Order")>
Public Property ObjectType() As Ektron.Cms.Common.EkEnumeration.CMSForumTypes
```

Forum.OrderBy Property

The order of forum search results. Choices are:

- **Title (see page 231)** - The content title (alphabetical)
- **ID** - The content ID number
- **Date Created** - The date the content was created
- **Date Modified** - The date the content was most recently modified
- **Editor** - The user who last edited the content (alphabetical)
- **Rank** - The rank assigned to the content. See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 725)" > "Display of Search (see page 474) Results" > "Search (see page 474) Result Ranking"

C#

```
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(5)]
[Description("Specify the order of the results. Choices are: Rank, Title, User, DateCreated,
DateModified. Default: Rank")]
[Personalizable(), WebBrowsable()]
public WebSearchOrder OrderBy;
```

Visual Basic

```
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(5)>
<Description("Specify the order of the results. Choices are: Rank, Title, User, DateCreated,
DateModified. Default: Rank")>
<Personalizable(), WebBrowsable()>
Public Property OrderBy() As WebSearchOrder
```

Forum.OrderDirection Property

Specify the order direction of forum search results. Choices are **Ascending** or **Descending**.

C#

```
[Browsable(true)]
[Category("Ektron")]
[TypeConverter(typeof(SortOrderTypeConverter))]
[Description("Specify the order direction of the results.")]
[Personalizable(), WebBrowsable()]
public string OrderDirection;
```

Visual Basic

```
<Browsable(true)>
<Category("Ektron")>
<TypeConverter(typeof(SortOrderTypeConverter))>
<Description("Specify the order direction of the results.")>
<Personalizable(), WebBrowsable()>
Public Property OrderDirection() As string
```

Forum.ProfileLink Property

Enter a link to the user profile page. This allows a user to click a link in the control and be taken to the user's profile page. There is a variables used within the link that represent the user's ID.

{0} - represents the user's ID.

The Web form can be relative or absolute. Below is an example.

```
userprofilepage.aspx?uid={0}
```

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue("Use Default"),
TypeConverter(typeof(ForumLinkConverter)), Description("The user profile link. {0} is the user
ID.") ]
public string ProfileLink;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue("Use Default"),
TypeConverter(typeof(ForumLinkConverter)), Description("The user profile link. {0} is the user
ID.") >
Public Property ProfileLink() As string
```

Forum.ProfileLinkTarget Property

Gets or Sets the target attribute for the user profile links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the user profile links.") ]
public ItemLinkTargets ProfileLinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the user profile links.") >
Public Property ProfileLinkTarget() As ItemLinkTargets
```

Forum.ShowCategories Property

If set to **True**, taxonomy categories will also be searched for the search text specified. The default is False.

True = Search (see page 474) across taxonomy categories

False = Do not search across taxonomy categories

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If this property is set to true, taxonomy categories will also be searched for
the search text specified.")]
public bool ShowCategories;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If this property is set to true, taxonomy categories will also be searched for
the search text specified.")>
Public Property ShowCategories() As bool
```

Forum.TaxCategories Property

Is server control in workarea

C#

```
[Browsable(false)]
public string TaxCategories;
```

Visual Basic

```
<Browsable(false)>
Public Property TaxCategories() As string
```

Forum.TaxonomyCols Property

Enter the number of columns in which this taxonomy/category will appear on the page.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)]
[Description("This aligns the taxonomy categories in number of columns.")]
public int TaxonomyCols;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)>
<Description("This aligns the taxonomy categories in number of columns.")>
Public Property TaxonomyCols() As Integer
```

Forum.TaxonomyMarkupLanguage Property

Enter name of a markup file that manages the display of the directory server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of the directory server control.")]
public string TaxonomyMarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of the directory server control.")>
Public Property TaxonomyMarkupLanguage() As string
```

Forum.Theme Property

Enter the custom theme's folder name. The theme can be relative to the site root or located in the current folder. If you do not specify a theme, the property uses the location defined by the Discussion Board's CSS theme property in the Workarea.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Enter the custom theme?s folder name. The theme can be relative to the site root
or located in the current folder. If you do not specify a theme, the property uses the
location defined by the Discussion Board?s CSS theme property in the Workarea.")]
[Personalizable(), WebBrowsable()]
public string Theme;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Enter the custom theme?s folder name. The theme can be relative to the site root
or located in the current folder. If you do not specify a theme, the property uses the
location defined by the Discussion Board?s CSS theme property in the Workarea.")>
<Personalizable(), WebBrowsable()>
Public Property Theme() As string
```

Friends Class

Summary description for Friends (Colleague Server Control API (see page 669))

Class Hierarchy

C#

```
[DefaultProperty("DefaultUserID"), ToolboxData("<{0}:Friends runat=server></{0}:Friends>")]
public class Friends : EkXsltWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultUserID"), ToolboxData("<{0}:Friends runat=server></{0}:Friends>")>
Public Class Friends
Inherits EkXsltWebPart
```

```
Inherits ICallbackEventHandler
```

File

Friends.cs

Friends Methods

The methods of the Friends class are listed here.

Public Methods

	Name	Description
	Fill (see page 280)	Populates the data for the Friends (see page 279).
	ToString (see page 280)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 280)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Friends.Fill Method

Populates the data for the Friends (see page 279).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Friends.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Friends.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

Friends Properties

The properties of the Friends class are listed here.

Public Properties

	Name	Description
	AvatarHeight (see page 281)	Gets or Sets the avatar height. 0 is unlimited.
	AvatarWidth (see page 282)	Gets or Sets the avatar width. 0 is unlimited.
	CacheInterval (see page 282)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DefaultUserID (see page 282)	The default user ID for this control to use when there is no matching dynamic parameter value passed.
	DisplayMode (see page 283)	Select the way this control displays colleagues. Choices are: **Directory (see page 196)** - this selection allows users to group their colleagues by folders. When this selection is chosen, the following menu items are added to the control: • **Add Folder (see page 683)** - allows a user to add a sub folder. • **Move** - allows a user to move users into a folder. This item only appears when there is a folder. **List** - shows a list colleagues in alphabetical order. **Pending** - shows a list pending colleagues. This list is made up of users who have sent you colleagues requests. **SentInvites** - shows a list... more (see page 283)
	DisplayXslt (see page 283)	Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. (optional).
	DynamicParameter (see page 283)	Gets or sets the Querystring parameter to read a user ID dynamically. To use the default user ID, leave blank.
	FriendLabel (see page 284)	Gets or sets the text for a colleague.
	FriendsLabel (see page 284)	Gets or Sets the text for colleagues. Similar to FriendLabel (see page 284), but is plural.
	Link (see page 284)	Enter a link to the user profile page. This allows a user to click a link in the Colleagues control and be taken to the user's profile page. There are two variables used within the link that represent the user's ID and the user's display name. **{0}** - represents the user's ID. **{1}** - represents the user's display name. You need to have both variables in the link. The Web form can be relative or absolute. Below is an example. `userprofilepage.aspx?uid={0}&dn={1}`
	LinkTarget (see page 285)	Gets or Sets the target attribute for the user links.
	MarkupLanguage (see page 285)	Enter name of a markup file that manages the display of this server control.
	MaxResults (see page 285)	Maximum number of results to be returned.
	ShowSortingOptions (see page 285)	Select wether the sorting options should be shown on the control. The following sorting options are available when this property is set to True: • **Colleagues** - a list of current colleagues. • **Pending Colleagues** - a list of colleagues who have sent you colleagues requests, but you have yet to accept. • **Invited** - a list of colleagues to whom you have sent colleague requests.
	Stylesheet (see page 286)	Specifies the Css to which this control will provide a link.
	XmlDoc (see page 286)	Returns the Xslt object for the collection.

Friends.AvatarHeight Property

Gets or Sets the avatar height. 0 is unlimited.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") ]
public int AvatarHeight;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") >
Public Property AvatarHeight() As Integer
```

Friends.AvatarWidth Property

Gets or Sets the avatar width. 0 is unlimited.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") ]
public int AvatarWidth;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") >
Public Property AvatarWidth() As Integer
```

Friends.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

Friends.DefaultUserID Property

The default user ID for this control to use when there is no matching dynamic parameter value passed.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The default user ID for this control to use when there is no matching dynamic
parameter value passed.")]
public long DefaultUserID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
```

```
<DefaultValue(0)>
<Description("The default user ID for this control to use when there is no matching dynamic
parameter value passed.")>
Public Property DefaultUserID() As long
```

Friends.DisplayMode Property

Select the way this control displays colleagues. Choices are:

Directory (see page 196) - this selection allows users to group their colleagues by folders. When this selection is chosen, the following menu items are added to the control:

- **Add Folder (see page 683)** - allows a user to add a sub folder.
- **Move** - allows a user to move users into a folder. This item only appears when there is a folder.

List - shows a list colleagues in alphabetical order.

Pending - shows a list pending colleagues. This list is made up of users who have sent you colleagues requests.

SentInvites - shows a list of users to whom you've sent colleague requests.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Directory), Description("This
property enables the user to show their colleagues either in the List, Pending, SentInvite or
Directory display mode.")]
public eDisplayMode DisplayMode;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Directory), Description("This
property enables the user to show their colleagues either in the List, Pending, SentInvite or
Directory display mode.")>
Public Property DisplayMode() As eDisplayMode
```

Friends.DisplayXslt Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. (optional).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control. (optional)")]
[TypeConverter(typeof(XsltFriendTypeConverter))]
[DefaultValue("(Path To Custom Xslt)")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control. (optional)")>
<TypeConverter(typeof(XsltFriendTypeConverter))>
<DefaultValue("(Path To Custom Xslt)")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

Friends.DynamicParameter Property

Gets or sets the Querystring parameter to read a user ID dynamically. To use the default user ID, leave blank.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
```

```
[Description("Gets or sets the Querystring parameter to read a user ID dynamically. To use the
default user ID, leave blank.")]
[Personalizable()]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Gets or sets the Querystring parameter to read a user ID dynamically. To use the
default user ID, leave blank.")>
<Personalizable()>
Public Property DynamicParameter() As string
```

Friends.FriendLabel Property

Gets or sets the text for a colleague.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue("Colleague"), Description("Gets or Sets the
text for a colleague.") ]
public string FriendLabel;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue("Colleague"), Description("Gets or Sets the
text for a colleague.") >
Public Property FriendLabel() As string
```

Friends.FriendsLabel Property

Gets or Sets the text for colleagues. Similar to FriendLabel (see page 284), but is plural.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue("Colleagues"), Description("Gets or Sets
the text for colleagues. Similar to FriendLabel, but is plural.") ]
public string FriendsLabel;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue("Colleagues"), Description("Gets or Sets
the text for colleagues. Similar to FriendLabel, but is plural.") >
Public Property FriendsLabel() As string
```

Friends.Link Property

Enter a link to the user profile page. This allows a user to click a link in the Colleagues control and be taken to the user's profile page. There are two variables used within the link that represent the user's ID and the user's display name.

{0} - represents the user's ID.

{1} - represents the user's display name.

You need to have both variables in the link. The Web form can be relative or absolute. Below is an example.

```
userprofilepage.aspx?uid={0}&dn={1}
```

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue("Use QuickLink"),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The user link. {0} is the user ID
and {1} is the user display name.") ]
public string Link;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue("Use QuickLink"),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The user link. {0} is the user ID
and {1} is the user display name.") >
Public Property Link() As string
```

Friends.LinkTarget Property

Gets or Sets the target attribute for the user links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the user links.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the user links.") >
Public Property LinkTarget() As ItemLinkTargets
```

Friends.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

Friends.MaxResults Property

Maximum number of results to be returned.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of users to fetch per page. If set to zero, there is no
limit.")]
[Personalizable()]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("The maximum number of users to fetch per page. If set to zero, there is no
limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

Friends.ShowSortingOptions Property

Select wether the sorting options should be shown on the control. The following sorting options are available when this property is set to True:

- **Colleagues** - a list of current colleagues.

- **Pending Colleagues** - a list of colleagues who have sent you colleagues requests, but you have yet to accept.
- **Invited** - a list of colleagues to whom you have sent colleague requests.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true), Description("Select wether the
sorting options should be shown on the control. The following sorting options are available
when this property is set to True: 'Colleagues' - a list of current colleagues. 'Pending
Colleagues' - a list of colleagues who have sent you colleague requests, but you have yet to
accept. 'Invited' - a list of colleagues to whom you have sent colleague requests.")]
public bool ShowSortingOptions;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true), Description("Select wether the
sorting options should be shown on the control. The following sorting options are available
when this property is set to True: 'Colleagues' - a list of current colleagues. 'Pending
Colleagues' - a list of colleagues who have sent you colleague requests, but you have yet to
accept. 'Invited' - a list of colleagues to whom you have sent colleague requests.")>
Public Property ShowSortingOptions() As bool
```

Friends.Stylesheet Property

Specifies the Css to which this control will provide a link.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control. Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control. Select either 'Use Default', 'None', or enter a
path to a custom Css file.")>
<Personalizable()>
Public Property Stylesheet() As string
```

Friends.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

HtmlEditDisplay Class

Class Hierarchy

Ektron.Cms.Controls.HtmlEditDisplay

C#

```
public class HtmlEditDisplay;
```

Visual Basic

```
Public Class HtmlEditDisplay
```

File

HtmlEditDisplay.cs

Description

This is class Ektron.Cms.Controls.HtmlEditDisplay.

HtmlEditDisplay Methods

The methods of the HtmlEditDisplay class are listed here.

Public Methods

	Name	Description
	BaseJavaScript (see page 287)	Create the Javascript that handles the onready event and other CMS specific setup functionality.
	BeginJavaScript (see page 288)	This returns the string which will surround all the JavaScript which is given in the CMSRender methods. This may contain any syntax, setup, includes, copyright message, or other options. This does not include the core JS files or the hidden field. Those are placed in by the eWebEditPro control.
	EndJavaScript (see page 288)	This will end the surrounding of the JavaScript created for the CMS.
S	MakeFieldNameFromId (see page 288)	This makes the names used by variable fields. Caution must be taken if this method is modified. The resulting names must match what the JavaScript is expecting for names. The JS is also assembling field names to set and retrieve data.
	RenderCMSFields (see page 288)	Creates the fields used when the content is posted. This creates fields which are posted to server with these IDs: • [ControlName]_Action = the action to take - save, check in, undo checkout • [ControlName]_ContentTitle = the content title • [ControlName]_ContentId = The ID of the content • [ControlName]_Language = The language of the content The content title caption is surrounded with a span that has this ID: • [ControlName]_TitleLabel All of the information for all the content information fields (except the content itself) is surrounded with a div tag with this ID: • [ControlName]_ContentInfo The valid values for the [ControlName]_Action value are: • Publish - Submit... more (see page 288)
	SelectedJavaScript (see page 289)	Renders any JavaScript based on options set by the developer through the properties.

HtmlEditDisplay.BaseJavaScript Method

Create the Javascript that handles the onready event and other CMS specific setup functionality.

C#

```
public string BaseJavaScript(string strTranslationFile, bool bShowCMSToolbar);
```

Visual Basic

```
Public Function BaseJavaScript(strTranslationFile As string, bShowCMSToolbar As bool) As string
```

Parameters

Parameters	Description
string strTranslationFile	Use m_strTranslationFile for this.

bool bShowCMSToolbar	Use m_bShowCMSToolbar for this.
iContentID	Use m_iContentId for this.
bIsLoggedIn	Use IsLoggedIn for this.

HtmlEditDisplay.BeginJavaScript Method

This returns the string which will surround all the JavaScript which is given in the CMSRender methods. This may contain any syntax, setup, includes, copyright message, or other options. This does not include the core JS files or the hidden field. Those are placed in by the eWebEditPro control.

C#

```
public string BeginJavaScript(long iContentId, int iLanguageID);
```

Visual Basic

```
Public Function BeginJavaScript(iContentId As long, iLanguageID As Integer) As string
```

Parameters

Parameters	Description
long iContentId	Use m_iContentId for this.
int iLanguageID	Use m_iLanguageID for this.
strWorkareaPath	Get using FindWorkareaLoc().

HtmlEditDisplay.EndJavaScript Method

This will end the surrounding of the JavaScript created for the CMS.

C#

```
public string EndJavaScript();
```

Visual Basic

```
Public Function EndJavaScript() As string
```

HtmlEditDisplay.MakeFieldNameFromId Method

This makes the names used by variable fields. Caution must be taken if this method is modified. The resulting names must match what the JavaScript is expecting for names. The JS is also assembling field names to set and retrieve data.

C#

```
public static string MakeFieldNameFromId(string strBaseId, string strSuffix);
```

Visual Basic

```
Public static Function MakeFieldNameFromId(strBaseId As string, strSuffix As string) As string
```

Parameters

Parameters	Description
string strBaseId	The name, or ID of the editor.
string strSuffix	The descriptive piece of the field name.

HtmlEditDisplay.RenderCMSFields Method

Creates the fields used when the content is posted. This creates fields which are posted to server with these IDs:

- [ControlName]_Action = the action to take - save, check in, undo checkout
- [ControlName]_ContentTitle = the content title
- [ControlName]_ContentId = The ID of the content
- [ControlName]_Language = The language of the content

The content title caption is surrounded with a span that has this ID:

- [ControlName]_TitleLabel

All of the information for all the content information fields (except the content itself) is surrounded with a div tag with this ID:

- [ControlName]_ContentInfo

The valid values for the [ControlName]_Action value are:

- Publish - Submit the posted content for publication
- CheckIn - Check in the content
- Save - Save the content and redisplay in the editor
- UndoCheckout - Undo the checkout of the content

C#

```
public string RenderCMSFields(string strContentTitle, string strTitleLabel, bool bShowTitle);
```

Visual Basic

```
Public Function RenderCMSFields(strContentTitle As string, strTitleLabel As string, bShowTitle
As bool) As string
```

Parameters

Parameters	Description
string strContentTitle	The title value text.
string strTitleLabel	Use RetrieveMessage(m_enumMessages.emsgContentTitle) for this.
bool bShowTitle	Use m_bShowTitle for this.

HtmlEditDisplay.SelectedJavaScript Method

Renders any JavaScript based on options set by the developer through the properties.

C#

```
public string SelectedJavaScript(int iSysDefaultLangID);
```

Visual Basic

```
Public Function SelectedJavaScript(iSysDefaultLangID As Integer) As string
```

HtmlEditor Class

This server control provides the ability to edit specific CMS content. The eWebEditPro editor can easily be placed onto a page for content editing. There is flexibility to control the toolbars, the available functionality, and the operations of the editor.

The control, at run time, produces the fields listed below. They are posted with the content and can be processed in the receiving page.

- [ControlName]_Action = the action to take - save, check in, undo checkout
- [ControlName]_ContentTitle = the content title
- [ControlName]_ContentId = the ID of the content
- [ControlName]_Language = the language of the content

The content title caption is surrounded with a span that has this ID: [ControlName]_TitleLabel.

All of the information for all the content information fields (except the content itself) is surrounded with a div tag with this ID: [ControlName]_ContentInfo.

The field values that are tied to the particular server control can be read through the respective properties:

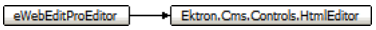
- [ControlName]_Action read with .ContentAction (see page 295)
- [ControlName]_ContentTitle read with .ContentTitle (see page 296)
- [ControlName]_ContentId read with .ContentId
- [ControlName]_Language read with .ContentLanguage
- [ControlName]_SafeEncode read with .SafeEncode (see page 301)

The content sent or posted is retrieved with the simple: .Text (see page 303)

The valid values for the [ControlName]_Action value are:

- Publish - Submit the posted content for publication
- CheckIn - Check in the content
- Save - Save the content and redisplay in the editor
- UndoCheckout - Undo the checkout of the content

Class Hierarchy



C#

```
[ DefaultProperty("ContentID"), ToolboxData("<{0}:HtmlEditor runat=server></{0}:HtmlEditor>"),
ToolboxBitmap("HtmlEditor.bmp"), Designer(typeof(HtmlEditor.HtmlEditDesigner)) ]
public class HtmlEditor : eWebEditProEditor;
```

Visual Basic

```
< DefaultProperty("ContentID"), ToolboxData("<{0}:HtmlEditor runat=server></{0}:HtmlEditor>"),
ToolboxBitmap("HtmlEditor.bmp"), Designer(typeof(HtmlEditor.HtmlEditDesigner)) >
Public Class HtmlEditor
Inherits eWebEditProEditor
```

File

HtmlEditor.cs

HtmlEditor.HtmlEditor Constructor

Constructor of the content editor. There are no parameters. The control depends on the settings given at design time or at run time.

C#

```
public HtmlEditor();
```

Visual Basic

```
Public Sub New()
```

HtmlEditor Methods

The methods of the HtmlEditor class are listed here.

Public Methods

	Name	Description
	ExecuteContentAction (see page 291)	This uses the ContentAction (see page 295) value and performs the specified action on the content contained within the server control's posted values. The content and the other property values are placed into the CMS400 according to the Action value. The possible action values are: • "Submit" • "CheckIn" • "Save" • "UndoCheckout" (cancel or undo)
	ToString (see page 291)	Returns the string which would be produced when the server control is rendered in the browser.
	WorkareaMode (see page 291)	Controls (see page 7) the operation of the internals of the server control. If this is not called, or given a value of 0, then this fills in all the necessary functionality to edit CMS content.

HtmlEditor.ExecuteContentAction Method

This uses the ContentAction (see page 295) value and performs the specified action on the content contained within the server control's posted values. The content and the other property values are placed into the CMS400 according to the Action value.

The possible action values are:

- "Submit"
- "CheckIn"
- "Save"
- "UndoCheckout" (cancel or undo)

C#

```
public string ExecuteContentAction();
```

Visual Basic

```
Public Function ExecuteContentAction() As string
```

Returns

HtmlEditor.ToString Method

Returns the string which would be produced when the server control is rendered in the browser.

C#

```
public new string ToString();
```

Visual Basic

```
Public new Function ToString() As string
```

Returns

The complete HTML produced by the control when rendered.

HtmlEditor.WorkareaMode Method

Controls (see page 7) the operation of the internals of the server control. If this is not called, or given a value of 0, then this fills in all the necessary functionality to edit CMS content.

C#

```
public void WorkareaMode(int iMode);
```

Visual Basic

```
Public Function WorkareaMode(iMode As Integer) As void
```

Parameters

Parameters	Description
int iMode	The mode to put the control into. It specifies the level of passiveness, where 0 denotes to add all required functionality automatically.

HtmlEditor Properties

The properties of the HtmlEditor class are listed here.

Public Properties

	Name	Description
	Archived (see page 293)	This signifies that the selected content is archived.
	Authenticated (see page 294)	DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and permits them to login through a login dialog in VisualStudio.NET.
	CmsSitePath (see page 294)	DESIGN TIME ONLY: Used to find the Web service required to access the CMS in Design Time.
	ContentAction (see page 295)	The action to perform on the posted content. The CMS buttons will set this value when selected. This is mostly useful on the receipt of the post as this will inform the code behind as to what action to take.
	ContentID (see page 295)	Specifies the content ID to load. This value must match a value that exists in the CMS database. The selection of content is similar to that in the ContentBlock (see page 148) control. The content can only be accessed if the user is logged in. This value can be set only on editor load, before the editor and contents are rendered. After editor load, this value is ignored. If this value is 0, then the code behind must handle making the calls that check out content and setting the content at edit time by placing it into the .Text (see page 303) property for the... more (see page 295)
	ContentTitle (see page 296)	The title for the selected Content (see page 822). This is set when new content is read. It can not be changed by the code behind. A text field is created, with the name EktContentTitle, that contains the description. The user can change it there.
	ContentType (see page 296)	This value describes the type of content to which the selected ID refers. If this value is blank, then the content is a custom type, to be defined type, or unsupported type. These are the possible types: • HTML • Form (see page 871)
	EditorReadyCall (see page 297)	This specifies which JavaScript function to run when the HTML editor is ready to start processing commands. This happens after the initialization of the editor and the loading of the initial content. This is the name, not the full function definition, of a JavaScript function. The function must follow the format defined in the eWebEditPro documentation for this type of method. Below is an example:
	EnableUpload (see page 297)	If 'true', the user can upload locally selected images and linked files that they have put into their content. If enabled, the automatic upload mechanism will allow uploading of the local files automatically with minimal user intervention. If 'false', the site developer must provide an upload page, such as the CMS library, where the upload can occur.
	IncludeContentInfo (see page 298)	If true, includes the information about the content in fields posted with a form post. (These fields are defined in the summary of this control.) Set the value to 'false' if the page maintains these values or the page is just using the editor to edit general formatted text. If the information must be posted, but not shown to the end user, then define the class [ControlName]_TitleLabel as having no display, or set the ShowTitle (see page 302) property to false.
	IsLoggedIn (see page 298)	DESIGN TIME ONLY: Read only property that lets a developer know if the user is currently log into the site as a user.
	Language (see page 298)	Specifies the ID of the language to use for the content to load. This value must match a value that exists in the CMS database.
	loggedInUserID (see page 299)	DESIGN TIME ONLY: Read only property that lets a developer know the user ID of the user who is currently logged in.
	loggedInUserName (see page 299)	DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currently logged in.

	Path (see page 300)	The path to the site that contains this control. This is not available in the properties window. It is meant for code behind which wants to override the path used by the CMS. The path is automatically determined by the control at run time, if it is not set by the code behind. This is only available from Code Behind.
	PreferredType (see page 300)	The display to use for viewing the content. This can be used to simplify the display or to debug the operation of editing content on the site. The options selectable are: • "activex" = Use the ActiveX control • "textarea" = Show HTML in text area The default is "activex". This is only available from Code Behind.
	SafeEncode (see page 301)	If true, then the posted HTML is encoded using URL encoding. This makes the transfer secure. If the content is encoded then it must be decoded on the receiving side. If the decode process is not wanted, then set this value to false. This will require the validateRequest attribute in the page definition be set to false.
	ShowCMSToolbar (see page 301)	Controls (see page 7) whether the CMS toolbar is displayed in the editor's toolbar. If this value is true, then commands containing CMS functionality are displayed in the editor's toolbar. The commands are for check-in, publish, and edit cancellation.
	ShowTitle (see page 302)	If 'true', the title of the selected content is displayed. The user can then edit the title. The submission of content posts up the title, even if it is not displayed to the user, unless the IncludeContentInfo (see page 298) property is set to a false value.
	StyleSheet (see page 302)	Assigns a single stylesheet to use with the edited content.
	Text (see page 303)	The HTML text being edited.
	ToolbarLevel (see page 303)	This specifies the level of the toolbar feature set offered to the user. It may be that they need a minimal set of functionality to keep the editing simple. It may be they need full functionality for a major edit. This defines the amount of functionality offered to the user. These are the value values: • "None" • "Reduced" • "Default" • "DataDesigner" • "DataEntry" • "Calendar (see page 76)" • "FormDesign" • "XSLTDesign" • "Full" • "Custom"
	ToolbarResetCall (see page 304)	This specifies which JavaScript function to run when the toolbar resets. The toolbar reset offers the opportunity for the script to affect the operation of the toolbar with actions such as adding or removing commands, disabling commands, or creating entirely new toolbars. This is the name, not the full function definition, of a JavaScript function. The function must follow the format defined in the eWebEditPro documentation for this type of method. Below is an example:
	TranslationFile (see page 304)	The translation file (or stream) that contains the XML elements for the translation of strings used in the editor. If this is blank, the client system's windows setting is used to select form the series of default translation files.

HtmlEditor.Archived Property

This signifies that the selected content is archived.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
```

```
[DefaultValue("")]
[Description("Is the content selected archived.")]
public bool Archived;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("Is the content selected archived.")>
Public Property Archived() As bool
```

HtmlEditor.Authenticated Property

DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and permits them to login through a login dialog in VisualStudio.NET.

C#

```
[Browsable(true)]
[DesignOnly(true)]
[Bindable(true)]
[Category("Ektron Authentication")]
[Editor(typeof(LoginUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(false)]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Description("DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and
permits them to login through a login dialog in VisualStudio.NET.")]
public string Authenticated;
```

Visual Basic

```
<Browsable(true)>
<DesignOnly(true)>
<Bindable(true)>
<Category("Ektron Authentication")>
<Editor(typeof(LoginUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(false)>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Description("DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and
permits them to login through a login dialog in VisualStudio.NET.")>
Public Property Authenticated() As string
```

HtmlEditor.CmsSitePath Property

DESIGN TIME ONLY: Used to find the Web service required to access the CMS in Design Time.

C#

```
[Browsable(true)]
[DesignOnly(true)]
[Bindable(true)]
[Category("Ektron Authentication")]
[DefaultValue("")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Editor(typeof(SelectWSUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Description("DESIGN TIME ONLY: Used to find the web service required to access the CMS in
Design Time.")]
public string CmsSitePath;
```

Visual Basic

```
<Browsable(true)>
<DesignOnly(true)>
<Bindable(true)>
```

```
<Category("Ektron Authentication")>
<DefaultValue("")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Editor(typeof(SelectWSUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("DESIGN TIME ONLY: Used to find the web service required to access the CMS in
Design Time.")>
Public Property CmsSitePath() As string
```

HtmlEditor.ContentAction Property

The action to perform on the posted content. The CMS buttons will set this value when selected. This is mostly useful on the receipt of the post as this will inform the code behind as to what action to take.

C#

```
[Bindable(true)]
[Browsable(false)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The action to perform with the content.")]
public string ContentAction;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The action to perform with the content.")>
Public Property ContentAction() As string
```

HtmlEditor.ContentID Property

Specifies the content ID to load. This value must match a value that exists in the CMS database. The selection of content is similar to that in the ContentBlock (see page 148) control. The content can only be accessed if the user is logged in. This value can be set only on editor load, before the editor and contents are rendered. After editor load, this value is ignored.

If this value is 0, then the code behind must handle making the calls that check out content and setting the content at edit time by placing it into the .Text (see page 303) property for the editor.

A hidden field, named EktContentID_[ControlName], is created to hold this value. It is posted up with the rest of the content.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(true)]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("The CMS ID of the content to load.")]
public long ContentID;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
```

```
<NotifyParentProperty(true)>
<ParenthesizePropertyName(true)>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("The CMS ID of the content to load.")>
Public Property ContentID() As long
```

HtmlEditor.ContentTitle Property

The title for the selected Content (see page 822). This is set when new content is read. It can not be changed by the code behind. A text field is created, with the name EktContentTitle, that contains the description. The user can change it there.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The title of the content.")]
public string ContentTitle;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The title of the content.")>
Public Property ContentTitle() As string
```

HtmlEditor.ContentType Property

This value describes the type of content to which the selected ID refers. If this value is blank, then the content is a custom type, to be defined type, or unsupported type. These are the possible types:

- HTML
- Form (see page 871)

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The type of content selected.")]
public string ContentType;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The type of content selected.")>
Public Property ContentType() As string
```

HtmlEditor.EditorReadyCall Property

This specifies which JavaScript function to run when the HTML editor is ready to start processing commands. This happens after the initialization of the editor and the loading of the initial content.

This is the name, not the full function definition, of a JavaScript function. The function must follow the format defined in the eWebEditPro documentation for this type of method. Below is an example:

```
function initEditorIsReady(sEditor, strURL, strAutoURL)
{ ... }
```

where the "initEditorIsReady" string is given to this property.

If this string is empty, the default CMS method is called.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("Specifies the JavaScript function to run when the editor is ready to start
processing commands.")]
public string EditorReadyCall;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("Specifies the JavaScript function to run when the editor is ready to start
processing commands.")>
Public Property EditorReadyCall() As string
```

HtmlEditor.EnableUpload Property

If 'true', the user can upload locally selected images and linked files that they have put into their content. If enabled, the automatic upload mechanism will allow uploading of the local files automatically with minimal user intervention. If 'false', the site developer must provide an upload page, such as the CMS library, where the upload can occur.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue(true)]
[Description("Allows the user to upload images and linked files.")]
public bool EnableUpload;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
```

```
<DefaultValue(true)>
<Description("Allows the user to upload images and linked files.")>
Public Property EnableUpload() As bool
```

HtmlEditor.IncludeContentInfo Property

If true, includes the information about the content in fields posted with a form post. (These fields are defined in the summary of this control.)

Set the value to 'false' if the page maintains these values or the page is just using the editor to edit general formatted text.

If the information must be posted, but not shown to the end user, then define the class [ControlName]_TitleLabel as having no display, or set the ShowTitle (see page 302) property to false.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue(true)]
[Description("Include the content information read from the database.")]
public bool IncludeContentInfo;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue(true)>
<Description("Include the content information read from the database.")>
Public Property IncludeContentInfo() As bool
```

HtmlEditor.IsLoggedIn Property

DESIGN TIME ONLY: Read only property that lets a developer know if the user is currently log into the site as a user.

C#

```
[Browsable(true)]
[DesignOnly(true)]
[Bindable(true)]
[Category("Ektron Authentication")]
[DefaultValue(false)]
[Description("DESIGN TIME ONLY: Read only property that displays whether the user is currently
logged into the CMS site.")]
public bool IsLoggedIn;
```

Visual Basic

```
<Browsable(true)>
<DesignOnly(true)>
<Bindable(true)>
<Category("Ektron Authentication")>
<DefaultValue(false)>
<Description("DESIGN TIME ONLY: Read only property that displays whether the user is currently
logged into the CMS site.")>
Public Property IsLoggedIn() As bool
```

HtmlEditor.Language Property

Specifies the ID of the language to use for the content to load. This value must match a value that exists in the CMS database.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(true)]
[DefaultValue(1033)]
[Description("The Lanugage of the content to load.")]
[Editor(typeof(SelectLanguageIDTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int Language;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(true)>
<DefaultValue(1033)>
<Description("The Lanugage of the content to load.")>
<Editor(typeof(SelectLanguageIDTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property Language() As Integer
```

HtmlEditor.loggedInUserID Property

DESIGN TIME ONLY: Read only property that lets a developer know the user ID of the user who is currently logged in.

C#

```
[Browsable(true)]
[DesignOnly(true)]
[Bindable(true)]
[Category("Ektron Authentication")]
[DefaultValue(0)]
[Description("DESIGN TIME ONLY: Read only property which displays the validated user ID for
the CMS site.")]
public long loggedInUserID;
```

Visual Basic

```
<Browsable(true)>
<DesignOnly(true)>
<Bindable(true)>
<Category("Ektron Authentication")>
<DefaultValue(0)>
<Description("DESIGN TIME ONLY: Read only property which displays the validated user ID for
the CMS site.")>
Public Property loggedInUserID() As long
```

HtmlEditor.loggedInUserName Property

DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currently logged in.

C#

```
[Browsable(true)]
[DesignOnly(true)]
[Bindable(true)]
[Category("Ektron Authentication")]
[DefaultValue("")]
[Description("DESIGN TIME ONLY: Read only property which displays the validated login name for
the CMS site.")]
public string loggedInUserName;
```

Visual Basic

```
<Browsable(true)>
<DesignOnly(true)>
<Bindable(true)>
<Category("Ektron Authentication")>
<DefaultValue("")>
<Description("DESIGN TIME ONLY: Read only property which displays the validated login name for
the CMS site.")>
Public Property loggedInUserName() As string
```

HtmlEditor.Path Property

The path to the site that contains this control. This is not available in the properties window. It is meant for code behind which wants to override the path used by the CMS.

The path is automatically determined by the control at run time, if it is not set by the code behind.

This is only available from Code Behind.

C#

```
[Bindable(true)]
[Browsable(false)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(true)]
[DefaultValue("")]
[Description("The path to the site that contains this control.")]
public new string Path;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(true)>
<DefaultValue("")>
<Description("The path to the site that contains this control.")>
Public new Property Path() As string
```

HtmlEditor.PreferredType Property

The display to use for viewing the content. This can be used to simplify the display or to debug the operation of editing content on the site.

The options selectable are:

- "activex" = Use the ActiveX control
- "textarea" = Show HTML in text area

The default is "activex". This is only available from Code Behind.

C#

```
[Bindable(true)]
[Browsable(false)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(true)]
[DefaultValue("")]
[Description("The path to the site that contains this control.")]
public new string PreferredType;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(true)>
<DefaultValue("")>
<Description("The path to the site that contains this control.")>
Public new Property PreferredType() As string
```

HtmlEditor.SafeEncode Property

If true, then the posted HTML is encoded using URL encoding. This makes the transfer secure. If the content is encoded then it must be decoded on the receiving side.

If the decode process is not wanted, then set this value to false. This will require the validateRequest attribute in the page definition be set to false.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue(true)]
[Description("Set to True to encode the HTML content.")]
public bool SafeEncode;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue(true)>
<Description("Set to True to encode the HTML content.")>
Public Property SafeEncode() As bool
```

Example 1

validateRequest="false"

Example 2

@Page Language (see page 298)="VB" validateRequest="false" AutoEventWireup="false" CodeFile="EditAnyHtmlSample.aspx.vb" Inherits="EditAnyHtmlSample"

HtmlEditor.ShowCMSToolbar Property

Controls (see page 7) whether the CMS toolbar is displayed in the editor's toolbar. If this value is true, then commands containing CMS functionality are displayed in the editor's toolbar. The commands are for check-in, publish, and edit cancellation.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue(true)]
```

```
[Description("Show CMS commands in the toolbar.")]
public bool ShowCMSToolbar;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue(true)>
<Description("Show CMS commands in the toolbar.")>
Public Property ShowCMSToolbar() As bool
```

HtmlEditor.ShowTitle Property

If 'true', the title of the selected content is displayed. The user can then edit the title. The submission of content posts up the title, even if it is not displayed to the user, unless the IncludeContentInfo (see page 298) property is set to a false value.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue(true)]
[Description("Set to True to show the content title for edit.")]
public bool ShowTitle;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue(true)>
<Description("Set to True to show the content title for edit.")>
Public Property ShowTitle() As bool
```

HtmlEditor.StyleSheet Property

Assigns a single stylesheet to use with the edited content.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The stylesheet to use for the edited content.")]
public new string StyleSheet;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
```

```
<Description("The stylesheet to use for the edited content.")>
Public new Property StyleSheet() As string
```

HtmlEditor.Text Property

The HTML text being edited.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The HTML text being edited.")]
public new string Text;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The HTML text being edited.")>
Public new Property Text() As string
```

HtmlEditor.ToolbarLevel Property

This specifies the level of the toolbar feature set offered to the user. It may be that they need a minimal set of functionality to keep the editing simple. It may be they need full functionality for a major edit. This defines the amount of functionality offered to the user.

These are the value values:

- "None"
- "Reduced"
- "Default"
- "DataDesigner"
- "DataEntry"
- "Calendar (see page 76)"
- "FormDesign"
- "XSLTDesign"
- "Full"
- "Custom"

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[TypeConverter(typeof(HtmlEditToolbarLvlTypeConverter))]
[DefaultValue("Default")]
[Description("Specifies the level of toolbar feature set offered to the user.")]
```

```
public string ToolbarLevel;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<TypeConverter(typeof(HtmlEditToolbarLvlTypeConverter))>
<DefaultValue("Default")>
<Description("Specifies the level of toolbar feature set offered to the user.")>
Public Property ToolbarLevel() As string
```

HtmlEditor.ToolbarResetCall Property

This specifies which JavaScript function to run when the toolbar resets. The toolbar reset offers the opportunity for the script to affect the operation of the toolbar with actions such as adding or removing commands, disabling commands, or creating entirely new toolbars.

This is the name, not the full function definition, of a JavaScript function. The function must follow the format defined in the eWebEditPro documentation for this type of method. Below is an example:

```
function initToolbarCommands(sEditor, strURL, strAutoURL)
{ ... }
```

Where the "initToolbarCommands" string is given to this property.

A toolbar reset is only called when the toolbar actually resets. This often happens when the editor starts, however, if the user has been given the opportunity to save their own toolbar setup then this will not be called on startup. It is always called when the user selects to reset the toolbar.

If this string is empty, the default CMS method is called.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("Specifies the JavaScript function to run when the toolbar is reset.")]
public string ToolbarResetCall;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("Specifies the JavaScript function to run when the toolbar is reset.")>
Public Property ToolbarResetCall() As string
```

HtmlEditor.TranslationFile Property

The translation file (or stream) that contains the XML elements for the translation of strings used in the editor. If this is blank, the client system's windows setting is used to select form the series of default translation files.

C#

```
[Bindable(true)]
```

```
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("What file to use for translations.")]
public string TranslationFile;
```

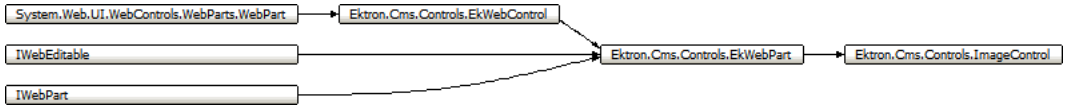
Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("What file to use for translations.")>
Public Property TranslationFile() As string
```

ImageControl Class

The ImageControl server control, when viewed on a Web form, displays an image stored within the CMS400.NET document management feature. When a user is logged in and has permission to edit the image, he can right click the image and select edit. This creates a drag and drop box that the user can use to update the file.

Class Hierarchy



C#

```
public class ImageControl : EkWebPart;
```

Visual Basic

```
Public Class ImageControl
Inherits EkWebPart
```

File

ImageControl.cs

ImageControl Methods

The methods of the ImageControl class are listed here.

Public Methods

	Name	Description
	Fill (see page 305)	Method that can be overridden and used to grab data from the back end for each server control.
	Wrap (see page 306)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

ImageControl.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

ImageControl.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

ImageControl Properties

The properties of the ImageControl class are listed here.

Public Properties

	Name	Description
	DefaultImageID (see page 306)	Gets or Sets the Default ImageID for this Control.
	DynamicParameter (see page 306)	Gets or Sets the QueryString parameter to read a Image ID dynamically. Set to String.Empty if you wish to always display the default Image ID (static.)
	FolderID (see page 307)	The ID of the folder you wish to select data from. Root is 0.
	TaxonomyID (see page 307)	The ID of the taxonomy where assets are added.
	Title (see page 308)	Gets or sets the Image Alternative/Title text. By default it is image file name.

ImageControl.DefaultImageID Property

Gets or Sets the Default ImageID for this Control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the Default Image ID for this Control.")]
[Personalizable(), WebBrowsable()]
public long DefaultImageID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the Default Image ID for this Control.")>
<Personalizable(), WebBrowsable()>
Public Property DefaultImageID() As long
```

ImageControl.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Image ID dynamically. Set to String.Empty if you wish to always display the

default Image ID (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Image ID dynamically. Set to
\"Use Default\" if you wish to always display the default Image ID (static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
[Personalizable(), WebBrowsable()]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Image ID dynamically. Set to
\"Use Default\" if you wish to always display the default Image ID (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
<Personalizable(), WebBrowsable()>
Public Property DynamicParameter() As string
```

ImageControl.FolderID Property

The ID of the folder you wish to select data from. Root is 0.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The ID of the folder you wish to select data from. Root is 0.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder you wish to select data from. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property FolderID() As long
```

ImageControl.TaxonomyID Property

The ID of the taxonomy where assets are added.

C#

```
[Category("Ektron")]
[DefaultValue("0")]
[Description("The ID of the taxonomy where assets are added.")]
[Personalizable(), WebBrowsable()]
public long TaxonomyID;
```

Visual Basic

```
<Category("Ektron")>
<DefaultValue("0")>
<Description("The ID of the taxonomy where assets are added.")>
<Personalizable(), WebBrowsable()>
Public Property TaxonomyID() As long
```

ImageControl.Title Property

Gets or sets the Image Alternative/Title text. By default it is image file name.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the Image Alternative/Title text. By defaut it is image file
name.")]
[DefaultValue("")]
public new string Title;
```

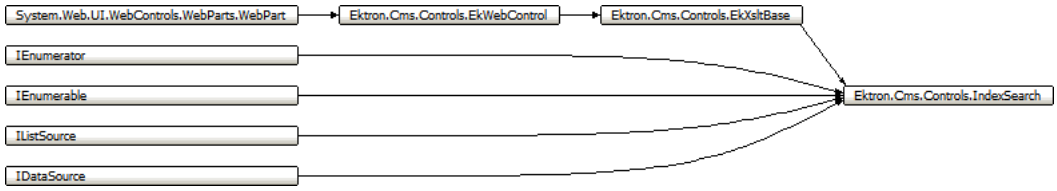
Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the Image Alternative/Title text. By defaut it is image file
name.")>
<DefaultValue("")>
Public new Property Title() As string
```

IndexSearch Class

Use the IndexSearch server control to display a Smart Form (see page 871)'s search screen on any Web form. This lets a user search your site based on XML parameters.

Class Hierarchy



C#

```
[DefaultProperty("XmlConfigId"), ToolboxData("<{0}:IndexSearch
runat=server></{0}:IndexSearch>"), PersistChildren(true)]
public class IndexSearch : EkXsltBase, IEnumerator, IEnumerable, IListSource, IDataSource;
```

Visual Basic

```
<DefaultProperty("XmlConfigId"), ToolboxData("<{0}:IndexSearch
runat=server></{0}:IndexSearch>"), PersistChildren(true)>
Public Class IndexSearch
Inherits EkXsltBase
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
```

File

IndexSearch.cs

IndexSearch.IndexSearch Constructor

Constructor for IndexSearch class.

C#

```
public IndexSearch();
```

Visual Basic

```
Public Sub New()
```

IndexSearch Structures

The structures of the IndexSearch class are listed here.

Structures

	Name	Description
	SearchPram (see page 309)	parameter structure used for search.

Ektron.Cms.Controls.IndexSearch.SearchPram Structure

parameter structure used for search.

C#

```
public struct SearchPram {
}
```

Visual Basic

```
Public Structure SearchPram
End Structure
```

File

IndexSearch.cs

IndexSearch Methods

The methods of the IndexSearch class are listed here.

Public Methods

	Name	Description
	AddParm (see page 309)	This is the overview for the AddParm method overload.
	CopyTo (see page 310)	Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.
	Fill (see page 310)	Calls up the database and updates the contents of the collection if the data is out of sync with the specified parameters.
	GetEnumerator (see page 311)	Gets a System.Collections.IEnumerator for the collection.
	GetSearchParmDataType (see page 311)	Returns the search parameter data type.
	GetSearchParmSearchType (see page 311)	Returns the search parameter search type.
	GetXPathValues (see page 311)	Returns array of strings using Xpaths.
	MoveNext (see page 312)	Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.
	Reset (see page 312)	Brings the index of the collection back to zero. Does not clear the collection.
	Search (see page 312)	This is the overview for the Search method overload.
	this (see page 313)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.
	Wrap (see page 313)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some HTML text.

AddParm Method

This is the overview for the AddParm method overload.

Overload List

	Name	Description
	IndexSearch.AddParm () (see page 310)	Adds a parameter to the search.
	IndexSearch.AddParm (SearchPram) (see page 310)	Adds parameter.

IndexSearch.AddParm Method ()

Adds a parameter to the search.

C#

```
public bool AddParm();
```

Visual Basic

```
Public Function AddParm() As bool
```

IndexSearch.AddParm Method (SearchPram)

Adds parameter.

C#

```
public bool AddParm(SearchPram Parm);
```

Visual Basic

```
Public Function AddParm(Parm As SearchPram) As bool
```

Parameters

Parameters	Description
SearchPram Parm	param name as SearchPram (see page 309)

Returns

True or False

IndexSearch.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

C#

```
public void CopyTo(Array array, int index);
```

Visual Basic

```
Public Function CopyTo(array As Array, index As Integer) As void
```

IndexSearch.Fill Method

Calls up the database and updates the contents of the collection if the data is out of sync with the specified parameters.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

IndexSearch.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

```
public IEnumerator GetEnumerator();
```

Visual Basic

```
Public Function GetEnumerator() As IEnumerator
```

IndexSearch.GetSearchParmDataType Method

Returns the search parameter data type.

C#

```
public EkEnumeration.XMLDataType GetSearchParmDataType(string val);
```

Visual Basic

```
Public Function GetSearchParmDataType(val As string) As EkEnumeration.XMLDataType
```

Parameters

Parameters	Description
string val	parameter name

Returns

parameter type

IndexSearch.GetSearchParmSearchType Method

Returns the search parameter search type.

C#

```
public EkEnumeration.XMLSearchRangeType GetSearchParmSearchType(string val);
```

Visual Basic

```
Public Function GetSearchParmSearchType(val As string) As EkEnumeration.XMLSearchRangeType
```

Parameters

Parameters	Description
string val	search type name

Returns

type name

IndexSearch.GetXPathValues Method

Returns array of strings using Xpaths.

C#

```
public string[] GetXPathValues(string XPath);
```

Visual Basic

```
Public Function GetXPathValues(XPath As string) As string()
```

Parameters

Parameters	Description
string XPath	XPath string

Returns

Array of Values

IndexSearch.MoveNext Method

Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.

C#

```
public bool MoveNext();
```

Visual Basic

```
Public Function MoveNext() As bool
```

Returns

Boolean indicating if the move was within the range of the array.

IndexSearch.Reset Method

Brings the index of the collection back to zero. Does not clear the collection.

C#

```
public void Reset();
```

Visual Basic

```
Public Function Reset() As void
```

Search Method

This is the overview for the Search method overload.

Overload List

	Name	Description
	IndexSearch.Search () (see page 312)	Performs the search action.
	IndexSearch.Search (System.Web.HttpRequest) (see page 312)	Search using request object.

IndexSearch.Search Method ()

Performs the search action.

C#

```
public void Search();
```

Visual Basic

```
Public Function Search() As void
```

IndexSearch.Search Method (System.Web.HttpRequest)

Search using request object.

C#

```
public void Search(System.Web.HttpRequest Request);
```

Visual Basic

```
Public Function Search(Request As System.Web.HttpRequest) As void
```

Parameters

Parameters	Description
System.Web.HttpRequest Request	Http Request

IndexSearch.this Indexer

Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.

C#

```
public Ektron.Cms.Controls.IndexSearchItem.ContentRow this[int i];
```

Visual Basic

```
Public Sub this(i As Integer)
```

IndexSearch.Wrap Method

Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some HTML text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

IndexSearch Properties

The properties of the IndexSearch class are listed here.

Public Properties

	Name	Description
	ButtonImgSrc (see page 314)	If you want to display an image on the submit button, enter the server path to it. (Optional)
	ButtonText (see page 315)	The text that appears on the Search (see page 312) button.
	ContentParameter (see page 315)	Checks the QueryString for this value and replaces the search results with that content block when specified. Leave blank to always display the search results.
	Count (see page 315)	Indicates the number of elements in your Collection (see page 95).
	Current (see page 316)	Gets the current object in the collection.
	CustomOrderBy (see page 316)	Order search results by an indexed field in the smart form. Set this property to the XPATH of that indexed field. If invalid XPATH is specified, search result ordering defaults to Rank.
	EkItems (see page 316)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 95). Holds more information than the exposed strongly typed this[i] index operator.
	EmptyResultMsg (see page 316)	The message that is displayed when the search returns no hits. (Optional)
	EncodeContentHtml (see page 316)	Set the property to true if you wish to html encode the Content (see page 822) Html in returned result."
	FolderId (see page 317)	The ID of the folder where to start the search. Root starts at "0".
	FormName (see page 317)	The form name.

	IsSynchronized (see page 317)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	Labelafter (see page 318)	Specify a label for the date field where search criteria is "After".
	Labelbefore (see page 318)	Specify a label for the date field where search criteria is "Before".
	LabelBetween (see page 318)	Specify label for numeric field where search criteria is based on range of values.
	LabelContains (see page 319)	Specify the label for a string field where search criteria "Contains".
	LabeldateBetween (see page 319)	Specify a label for the date field where search criteria based on a range of dates.
	LabelEqual (see page 319)	Specify label for numeric field where search criteria is "Equal".
	LabelExactPhrase (see page 320)	Specify a label for the string field where search criteria "Exact Phrase".
	LabelGreaterThan (see page 320)	Specify label for numeric field where search criteria is "Greater Than".
	Labelinputdate (see page 320)	Specify a label for the message that appears when date entered are blank/incorrect.
	Labelinputnumber (see page 320)	Specify a label for the message that appears when values entered are blank/incorrect.
	LabelLessThan (see page 321)	Specify label for numeric field where search criteria "Less Than".
	LabelOn (see page 321)	Specify a label for the Date field where search criteria is "On".
	LabelSelect (see page 321)	Specify labels for first item on selection dropdown.
	LinkTarget (see page 322)	Gets or sets the target window for the result to be displayed.
	MaxResults (see page 322)	The total number of results to display. If set to zero, unlimited results will be set.
	OrderBy (see page 322)	Specify the order of the results. Choices are: • **Title** - the title of the content block • **ID** - the XML content block ID number • **Date Created** - the date the content block was created • **Date Modified** - the date the content block was last modified • **AuthorLName** - the last author's last name • **AuthorFname** - the last author's first name • **StartDate** - the GoLive date
	OrderByDirection (see page 323)	The direction the results will be sorted in. This property works with the OrderBy (see page 322) property only. Choices are: • Ascending • Descending
	Recursive (see page 323)	Select True if you wish to search the subfolders of the selected FolderPath.
	SearchParmXML (see page 323)	Predefine a set of search criteria.
	ShowSearchBoxAlways (see page 324)	If this property is set to False, the search box will not be visible on PostBack.
	SyncRoot (see page 324)	Gets an object that can be used to synchronize the collection.
	Weighted (see page 324)	Select True if you wish perform a weighted search.
	XmlConfigId (see page 324)	ID of the Smart Form (see page 871) whose indexed fields will appear on the search screen.
	XmlDoc (see page 325)	Returns the XSLT object for the collection.

IndexSearch.ButtonImgSrc Property

If you want to display an image on the submit button, enter the server path to it. (Optional)

C#

```
[Bindable(true)]
[Category("Ektron")]
```

```
[DefaultValue("")]
[Description("Optional: The path to the image you wish to use for the Search button.")]
public string ButtonImgSrc;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Optional: The path to the image you wish to use for the Search button.")>
Public Property ButtonImgSrc() As string
```

IndexSearch.ButtonText Property

The text that appears on the Search (see page 312) button.

C#

```
[Category("Ektron")]
[Description("The text of the Search button.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_BTN)]
public string ButtonText;
```

Visual Basic

```
<Category("Ektron")>
<Description("The text of the Search button.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_BTN)>
Public Property ButtonText() As string
```

IndexSearch.ContentParameter Property

Checks the QueryString for this value and replaces the search results with that content block when specified. Leave blank to always display the search results.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and replaces the search results with a
content block when specified. Leave blank to always display the search results.")]
public string ContentParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and replaces the search results with a
content block when specified. Leave blank to always display the search results.")>
Public Property ContentParameter() As string
```

IndexSearch.Count Property

Indicates the number of elements in your Collection (see page 95).

C#

```
[Category("Data")]
[Description("Indicates the number of elements in your Collection.")]
public int Count;
```

Visual Basic

```
<Category("Data")>
```

```
<Description("Indicates the number of elements in your Collection.")>
Public Property Count() As Integer
```

IndexSearch.Current Property

Gets the current object in the collection.

C#

```
[Browsable(false)]
public object Current;
```

Visual Basic

```
<Browsable(false)>
Public Property Current() As object
```

IndexSearch.CustomOrderBy Property

Order search results by an indexed field in the smart form. Set this property to the XPATH of that indexed field. If invalid XPATH is specified, search result ordering defaults to Rank.

C#

```
[Category("Ektron"), DefaultValue(""), Description("Order search results by an indexed field
in the smart form. Set this property to the XPATH of that indexed field.")]
public string CustomOrderBy;
```

Visual Basic

```
<Category("Ektron"), DefaultValue(""), Description("Order search results by an indexed field
in the smart form. Set this property to the XPATH of that indexed field.")>
Public Property CustomOrderBy() As string
```

IndexSearch.EkItems Property

Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 95). Holds more information than the exposed strongly typed this[i] index operator.

C#

```
public Ektron.Cms.Common.IndexSearchItems EkItems;
```

Visual Basic

```
Public Property EkItems() As Ektron.Cms.Common.IndexSearchItems
```

IndexSearch.EmptyResultMsg Property

The message that is displayed when the search returns no hits. (Optional)

C#

```
[Category("Ektron")]
[Description("Optional: The return message if the search returns no hits.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_NOHITS)]
public string EmptyResultMsg;
```

Visual Basic

```
<Category("Ektron")>
<Description("Optional: The return message if the search returns no hits.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_NOHITS)>
Public Property EmptyResultMsg() As string
```

IndexSearch.EncodeContentHtml Property

Set the property to true if you wish to html encode the Content (see page 822) Html in returned result."

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to true, Content Html in returned result will be
encoded.")]
public bool EncodeContentHtml;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to true, Content Html in returned result will be
encoded.")>
Public Property EncodeContentHtml() As bool
```

IndexSearch.FolderId Property

The ID of the folder where to start the search. Root starts at "0".

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("The ID of the folder you wish to start the search. Root starts at \"0\".")]
[Editor(typeof(GetFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long FolderId;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The ID of the folder you wish to start the search. Root starts at \"0\".")>
<Editor(typeof(GetFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderId() As long
```

IndexSearch.FormName Property

The form name.

C#

```
[Bindable(false), Browsable(false)]
[Description("")]
public string FormName;
```

Visual Basic

```
<Bindable(false), Browsable(false)>
<Description("")>
Public Property FormName() As string
```

IndexSearch.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

```
[Browsable(false)]
public bool IsSynchronized;
```

Visual Basic

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

IndexSearch.Labelafter Property

Specify a label for the date field where search criteria is "After".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify a label for the date field where search criteria is \"After\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_AFTER)]
public string Labelafter;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify a label for the date field where search criteria is \"After\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_AFTER)>
Public Property Labelafter() As string
```

IndexSearch.Labelbefore Property

Specify a label for the date field where search criteria is "Before".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify a label for the date field where search criteria is \"Before\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_BFORE)]
public string Labelbefore;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify a label for the date field where search criteria is \"Before\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_BFORE)>
Public Property Labelbefore() As string
```

IndexSearch.LabelBetween Property

Specify label for numeric field where search criteria is based on range of values.

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for numeric field where search criteria is based on range of
values.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_BTWEEN)]
public string LabelBetween;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for numeric field where search criteria is based on range of
values.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_BTWEEN)>
Public Property LabelBetween() As string
```

IndexSearch.LabelContains Property

Specify the label for a string field where search criteria "Contains".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify the label for a string field where search criteria \"Contains\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_CONTS)]
public string LabelContains;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify the label for a string field where search criteria \"Contains\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_CONTS)>
Public Property LabelContains() As string
```

IndexSearch.LabeldateBetween Property

Specify a label for the date field where search criteria based on a range of dates.

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify a label for the date field where search criteria based on a range of
dates.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_DBTWEEN)]
public string LabeldateBetween;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify a label for the date field where search criteria based on a range of
dates.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_DBTWEEN)>
Public Property LabeldateBetween() As string
```

IndexSearch.LabelEqual Property

Specify label for numeric field where search criteria is "Equal".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for numeric field where search criteria is \"Equal\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_EQUAL)]
public string LabelEqual;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for numeric field where search criteria is \"Equal\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_EQUAL)>
Public Property LabelEqual() As string
```

IndexSearch.LabelExactPhrase Property

Specify a label for the string field where search criteria "Exact Phrase".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify a label for the string field where search criteria \"Exact Phrase\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_XACT)]
public string LabelExactPhrase;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify a label for the string field where search criteria \"Exact Phrase\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_XACT)>
Public Property LabelExactPhrase() As string
```

IndexSearch.LabelGreaterThan Property

Specify label for numeric field where search criteria is "Greater Than".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for numeric field where search criteria is \"Greater Than\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_GRTER)]
public string LabelGreaterThan;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for numeric field where search criteria is \"Greater Than\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_GRTER)>
Public Property LabelGreaterThan() As string
```

IndexSearch.Labelinputdate Property

Specify a label for the message that appears when date entered are blank/incorrect.

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for message that appears when date entered are blank/incorrect.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_IPUTDT)]
public string Labelinputdate;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for message that appears when date entered are blank/incorrect.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_IPUTDT)>
Public Property Labelinputdate() As string
```

IndexSearch.Labelinputnumber Property

Specify a label for the message that appears when values entered are blank/incorrect.

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify a label for the message that appears when values entered are
blank/incorrect.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_IPUTNUM)]
public string Labelinputnumber;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify a label for the message that appears when values entered are
blank/incorrect.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_IPUTNUM)>
Public Property Labelinputnumber() As string
```

IndexSearch.LabelLessThan Property

Specify label for numeric field where search criteria "Less Than".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for numeric field where search criteria is \"Less Than\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_LESS)]
public string LabelLessThan;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for numeric field where search criteria is \"Less Than\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_LESS)>
Public Property LabelLessThan() As string
```

IndexSearch.LabelOn Property

Specify a label for the Date field where search criteria is "On".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for Date field where search criteria is \"On\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_ON)]
public string LabelOn;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for Date field where search criteria is \"On\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_ON)>
Public Property LabelOn() As string
```

IndexSearch.LabelSelect Property

Specify labels for first item on selection dropdown.

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify labels for first item on selection dropdown.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_SELECT)]
public string LabelSelect;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify labels for first item on selection dropdown.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_SELECT)>
Public Property LabelSelect() As string
```

IndexSearch.LinkTarget Property

Gets or sets the target window for the result to be displayed.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

IndexSearch.MaxResults Property

The total number of results to display. If set to zero, unlimited results will be set.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The total number of results to display. If set to zero, unlimited results will
be set.")]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The total number of results to display. If set to zero, unlimited results will
be set.")>
Public Property MaxResults() As Integer
```

IndexSearch.OrderBy Property

Specify the order of the results. Choices are:

- **Title** - the title of the content block
- **ID** - the XML content block ID number
- **Date Created** - the date the content block was created
- **Date Modified** - the date the content block was last modified
- **AuthorLName** - the last author's last name

- **AuthorFname** - the last author's first name
- **StartDate** - the GoLive date

C#

```
[Category("Ektron"), DefaultValue("Title"), Description("Specify the order of the results.
Choices are: Title, ID, Date Created, Date Modified, AuthorLName, AuthorFName or StartDate")]
public string OrderBy;
```

Visual Basic

```
<Category("Ektron"), DefaultValue("Title"), Description("Specify the order of the results.
Choices are: Title, ID, Date Created, Date Modified, AuthorLName, AuthorFName or StartDate")>
Public Property OrderBy() As string
```

IndexSearch.OrderByDirection Property

The direction the results will be sorted in. This property works with the OrderBy (see page 322) property only. Choices are:

- Ascending
- Descending

C#

```
[Category("Ektron"), TypeConverter(typeof(SortOrderTypeConverter)), Description("Specify the
order direction of the results.")]
public string OrderByDirection;
```

Visual Basic

```
<Category("Ektron"), TypeConverter(typeof(SortOrderTypeConverter)), Description("Specify the
order direction of the results.")>
Public Property OrderByDirection() As string
```

IndexSearch.Recursive Property

Select True if you wish to search the subfolders of the selected FolderPath.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Select True if you wish to search the subfolders of the selected FolderPath.")]
public bool Recursive;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Select True if you wish to search the subfolders of the selected FolderPath.")>
Public Property Recursive() As bool
```

IndexSearch.SearchParmXML Property

Predefine a set of search criteria.

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Predefine a search criteria.")]
[Editor(typeof(SearchParmColUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public string SearchParmXML;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
```

```
<Description("Predefine a search criteria.")>
<Editor(typeof(SearchParmColUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property SearchParmXML() As string
```

IndexSearch.ShowSearchBoxAlways Property

If this property is set to False, the search box will not be visible on PostBack.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to False, the search box will not be visible on
PostBack.")]
public bool ShowSearchBoxAlways;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to False, the search box will not be visible on
PostBack.")>
Public Property ShowSearchBoxAlways() As bool
```

IndexSearch.SyncRoot Property

Gets an object that can be used to synchronize the collection.

C#

```
[Browsable(false)]
public object SyncRoot;
```

Visual Basic

```
<Browsable(false)>
Public Property SyncRoot() As object
```

IndexSearch.Weighted Property

Select True if you wish perform a weighted search.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Select True if you wish to do an OR search between indexed fields.")]
public bool Weighted;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Select True if you wish to do an OR search between indexed fields.")>
Public Property Weighted() As bool
```

IndexSearch.XmlConfigId Property

ID of the Smart Form (see page 871) whose indexed fields will appear on the search screen.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("ID of the Smart Form whose indexed fields will appear on the search screen.")]
```

```
[Editor(typeof(XMLConfigIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long XmlConfigId;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("ID of the Smart Form whose indexed fields will appear on the search screen.")>
<Editor(typeof(XMLConfigIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property XmlConfigId() As long
```

IndexSearch.XmlDoc Property

Returns the XSLT object for the collection.

C#

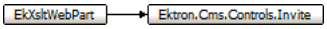
```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

Invite Class

Class Hierarchy



C#

```
public class Invite : EkXsltWebPart;
```

Visual Basic

```
Public Class Invite
Inherits EkXsltWebPart
```

File

Invite.cs

Invite Methods

The methods of the Invite class are listed here.

Public Methods

	Name	Description
	Fill (see page 325)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 326)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 326)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Invite.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Invite.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Invite.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

Invite Properties

The properties of the Invite class are listed here.

Public Properties

	Name	Description
	DisplayXslt (see page 326)	The Xslt to use to render the control. (optional)
	FriendMessageId (see page 327)	The ID of the message to send in the link section of the friend request email.
	InviteMessageId (see page 327)	The ID of the message to send in the link section of the invite email.
	OptionalMessageText (see page 327)	This is the default text that will be shown in the optional-message textarea, which can be edited by users.
	Stylesheet (see page 328)	This is the default stylesheet used to style the appearance of the control.
	XmlDoc (see page 328)	Returns the Xslt object for the collection.

Invite.DisplayXslt Property

The Xslt to use to render the control. (optional)

C#

```
[Category("Ektron")]
[Description("The Xslt to use to render the control. (optional)")]
[TypeConverter(typeof(XsltInviteTypeConverter))]
[DefaultValue("Basic Invite")]
public override string DisplayXslt;
```

Visual Basic

```
<Category("Ektron")>
<Description("The Xslt to use to render the control. (optional)")>
<TypeConverter(typeof(XsltInviteTypeConverter))>
<DefaultValue("Basic Invite")>
```

```
Public override Property DisplayXslt() As string
```

Invite.FriendMessageId Property

The ID of the message to send in the link section of the friend request email.

C#

```
[Category("Ektron")]
[Description("The ID of the message to send in the link section of the friend request email
(defined in the workarea), if zero then uses the default message.")]
[TypeConverter(typeof(int))]
[DefaultValue("0")]
public long FriendMessageId;
```

Visual Basic

```
<Category("Ektron")>
<Description("The ID of the message to send in the link section of the friend request email
(defined in the workarea), if zero then uses the default message.")>
<TypeConverter(typeof(int))>
<DefaultValue("0")>
Public Property FriendMessageId() As long
```

Invite.InviteMessageId Property

The ID of the message to send in the link section of the invite email.

C#

```
[Category("Ektron")]
[Description("The ID of the message to send in the link section of the invite email (defined
in the workarea), if zero then uses the default message.")]
[TypeConverter(typeof(int))]
[DefaultValue("0")]
public long InviteMessageId;
```

Visual Basic

```
<Category("Ektron")>
<Description("The ID of the message to send in the link section of the invite email (defined
in the workarea), if zero then uses the default message.")>
<TypeConverter(typeof(int))>
<DefaultValue("0")>
Public Property InviteMessageId() As long
```

Invite.OptionalMessageText Property

This is the default text that will be shown in the optional-message textarea, which can be edited by users.

C#

```
[Category("Ektron")]
[Description("This is the default text that will be shown in the optional-message textarea,
which can be edited by users.")]
[TypeConverter(typeof(string))]
[DefaultValue("Hello, I am sending you this email because I would like you to join me at this
website.")]
public string OptionalMessageText;
```

Visual Basic

```
<Category("Ektron")>
<Description("This is the default text that will be shown in the optional-message textarea,
which can be edited by users.")>
<TypeConverter(typeof(string))>
<DefaultValue("Hello, I am sending you this email because I would like you to join me at this
website.")>
Public Property OptionalMessageText() As string
```

Invite.Stylesheet Property

This is the default stylesheet used to style the appearance of the control.

C#

```
[Category("Ektron")]
[Description("This is the default stylesheet used to style the appearance of the control.")]
[TypeConverter(typeof(string))]
[DefaultValue("Community.css")]
public string Stylesheet;
```

Visual Basic

```
<Category("Ektron")>
<Description("This is the default stylesheet used to style the appearance of the control.")>
<TypeConverter(typeof(string))>
<DefaultValue("Community.css")>
Public Property Stylesheet() As string
```

Invite.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

LanguageAPI Class

Control that can be used to override the default system language assignment behavior.

Class Hierarchy

C#

```
public class LanguageAPI : WebControl;
```

Visual Basic

```
Public Class LanguageAPI
Inherits WebControl
```

File

LanguageAPI.cs

LanguageAPI.LanguageAPI Constructor

Upon initialization, SiteLanguage (see page 332) is set to DefaultLanguage (see page 331).

C#

```
public LanguageAPI();
```

Visual Basic

```
Public Sub New()
```

LanguageAPI Methods

The methods of the LanguageAPI class are listed here.

Public Methods

	Name	Description
	Fill (see page 329)	Use this function along with LanguageLogic to implement your custom language control logic.
	GetLanguage (see page 329)	Returns a Language string corresponding to a given LanguageID.
	GetLanguageID (see page 329)	Returns the LanguageID corresponding to a given Language string.
	IsValid (see page 329)	This is the overview for the IsValid method overload.

LanguageAPI.Fill Method

Use this function along with LanguageLogic to implement your custom language control logic.

C#

```
public virtual void Fill();
```

Visual Basic

```
Public virtual Function Fill() As void
```

LanguageAPI.GetLanguage Method

Returns a Language string corresponding to a given LanguageID.

C#

```
public string GetLanguage(int LanguageID);
```

Visual Basic

```
Public Function GetLanguage(LanguageID As Integer) As string
```

LanguageAPI.GetLanguageID Method

Returns the LanguageID corresponding to a given Language string.

C#

```
public int GetLanguageID(string Language);
```

Visual Basic

```
Public Function GetLanguageID(Language As string) As Integer
```

IsValid Method

This is the overview for the IsValid method overload.

Overload List

	Name	Description
	LanguageAPI.IsValid (int) (see page 329)	Returns true if the passed LanguageID is an active system language.
	LanguageAPI.IsValid (string) (see page 330)	Returns true if the passed Language is an active system language.

LanguageAPI.IsValid Method (int)

Returns true if the passed LanguageID is an active system language.

C#

```
public bool IsValid(int LanguageID);
```

Visual Basic

```
Public Function IsValid(LanguageID As Integer) As bool
```

LanguageAPI.IsValid Method (string)

Returns true if the passed Language is an active system language.

C#

```
public bool IsValid(string Language);
```

Visual Basic

```
Public Function IsValid(Language As string) As bool
```

LanguageAPI Properties

The properties of the LanguageAPI class are listed here.

Public Properties

	Name	Description
	CurrentLanguage (see page 330)	Returns the current language (from a cookie).
	CurrentLanguageID (see page 330)	Returns the current language ID (from a cookie).
	DefaultLanguage (see page 331)	If not overridden, the system string "Default" is used.
	DefaultLanguageID (see page 331)	If not overridden, the ID '0' is used.
	LangaugeIdList (see page 331)	Returns a list of all the language IDs that are activated in Ektron CMS400.NET
	LangaugeTitleList (see page 331)	Returns a list of all the languages that are activated in Ektron CMS400.NET.
	MultiLanguageEnabled (see page 331)	Tells if the site supports multi-language mode. • **True** = Multi-language enabled • **False** = Multi-language not enabled
	OverrideDefault (see page 332)	If you wish to override the default behavior of CMS400.NET, set this to true and implement your logic in the 'LanguageLogic()' overridable method. Currently this property returns the set value only if SiteLanguage (see page 332) is not equal to DefaultLanguage (see page 331). If they are equal, false is returned.
	SiteLanguage (see page 332)	Sets the site language of the site. Runs through the IsValid (see page 329) function to verify that the language is an active language in the system.
	SiteLanguageID (see page 332)	If not overridden, the ID '0' is used.

LanguageAPI.CurrentLanguage Property

Returns the current language (from a cookie).

C#

```
[Browsable(false)]
public string CurrentLanguage;
```

Visual Basic

```
<Browsable(false)>
Public Property CurrentLanguage() As string
```

LanguageAPI.CurrentLanguageID Property

Returns the current language ID (from a cookie).

C#

```
[Browsable(false)]
public int CurrentLanguageID;
```

Visual Basic

```
<Browsable(false)>
Public Property CurrentLanguageID() As Integer
```

LanguageAPI.DefaultLanguage Property

If not overridden, the system string "Default" is used.

C#

```
[Browsable(false)]
public string DefaultLanguage;
```

Visual Basic

```
<Browsable(false)>
Public Property DefaultLanguage() As string
```

LanguageAPI.DefaultLanguageID Property

If not overridden, the ID '0' is used.

C#

```
[Browsable(false)]
public int DefaultLanguageID;
```

Visual Basic

```
<Browsable(false)>
Public Property DefaultLanguageID() As Integer
```

LanguageAPI.LangaugeIdList Property

Returns a list of all the language IDs that are activated in Ektron CMS400.NET

C#

```
[Browsable(false)]
public int LangaugeIdList;
```

Visual Basic

```
<Browsable(false)>
Public Property LangaugeIdList() As Integer
```

LanguageAPI.LangaugeTitleList Property

Returns a list of all the languages that are activated in Ektron CMS400.NET.

C#

```
[Browsable(false)]
public string LangaugeTitleList;
```

Visual Basic

```
<Browsable(false)>
Public Property LangaugeTitleList() As string
```

LanguageAPI.MultiLanguageEnabled Property

Tells if the site supports multi-language mode.

- **True** = Multi-language enabled

- **False** = Multi-language not enabled

C#

```
[Browsable(false)]
public bool MultiLanguageEnabled;
```

Visual Basic

```
<Browsable(false)>
Public Property MultiLanguageEnabled() As bool
```

LanguageAPI.OverrideDefault Property

If you wish to override the default behavior of CMS400.NET, set this to true and implement your logic in the 'LanguageLogic()' overridable method. Currently this property returns the set value only if SiteLanguage (see page 332) is not equal to DefaultLanguage (see page 331). If they are equal, false is returned.

C#

```
[Browsable(false)]
public bool OverrideDefault;
```

Visual Basic

```
<Browsable(false)>
Public Property OverrideDefault() As bool
```

LanguageAPI.SiteLanguage Property

Sets the site language of the site. Runs through the IsValid (see page 329) function to verify that the language is an active language in the system.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the site language of the site. Runs through the IsValid function to verify
that the language is an active language in the system.")]
[TypeConverter(typeof(LanguageTypeConverter))]
public string SiteLanguage;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the site language of the site. Runs through the IsValid function to verify
that the language is an active language in the system.")>
<TypeConverter(typeof(LanguageTypeConverter))>
Public Property SiteLanguage() As string
```

LanguageAPI.SiteLanguageID Property

If not overridden, the ID '0' is used.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the site language of the site. Runs through the IsValid function to verify
that the language is an active language in the system.")]
public int SiteLanguageID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the site language of the site. Runs through the IsValid function to verify
that the language is an active language in the system.")>
Public Property SiteLanguageID() As Integer
```

LanguageSelect Class

The LanguageSelect server control displays a language selection dropdown list on an Ektron CMS400.NET Web page. The dropdown box lists all languages selected on the Settings > Configuration > Language (see page 227) settings screen.

Class Hierarchy

C#

```
public class LanguageSelect : EkWebControl;
```

Visual Basic

```
Public Class LanguageSelect
Inherits EkWebControl
```

File

LanguageSelect.cs

LanguageSelect Methods

The methods of the LanguageSelect class are listed here.

Public Methods

	Name	Description
	Fill (see page 333)	Method that can be overridden and used to grab data from the back end for each server control.
	Wrap (see page 333)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

LanguageSelect.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

LanguageSelect.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

Html string wrapped with a toolbar.

LanguageSelect Properties

The properties of the LanguageSelect class are listed here.

Public Properties

	Name	Description
	LabelName (see page 334)	Gets or sets the label next to the language select dropdown box.
	Text (see page 334)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.
	WrapTag (see page 334)	Allows a developer to specify control's tag. Default is assumed "".

LanguageSelect.LabelName Property

Gets or sets the label next to the language select dropdown box.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Label next to the language select. The default value is set to
\"res_lngsel_lbl\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_LANGSEL_LBL)]
public string LabelName;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Label next to the language select. The default value is set to
\"res_lngsel_lbl\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_LANGSEL_LBL)>
Public Property LabelName() As string
```

LanguageSelect.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

```
public override string Text;
```

Visual Basic

```
Public override Property Text() As string
```

LanguageSelect.WrapTag Property

Allows a developer to specify control's tag. Default is assumed "".

C#

```
[Bindable(false)]
[Browsable(false)]
[DesignerSerializationVisibility(DesignerSerializationVisibility.Hidden)]
public override string WrapTag;
```

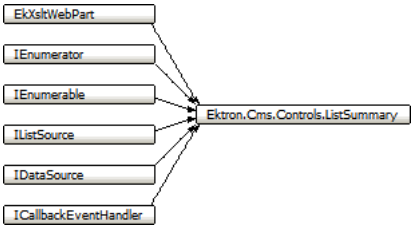
Visual Basic

```
<Bindable(false)>
<Browsable(false)>
<DesignerSerializationVisibility(DesignerSerializationVisibility.Hidden)>
Public override Property WrapTag() As string
```

ListSummary Class

Summary description for ListSummary.

Class Hierarchy



C#

```
[DefaultProperty("FolderID"), ToolboxData("<{0}:ListSummary runat=server></{0}:ListSummary>")]
public class ListSummary : EkXsltWebPart, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("FolderID"), ToolboxData("<{0}:ListSummary runat=server></{0}:ListSummary>")>
Public Class ListSummary
Inherits EkXsltWebPart
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
Inherits ICallbackEventHandler
```

File

ListSummary.cs

ListSummary Methods

The methods of the ListSummary class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 335)	Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.
	Fill (see page 336)	Populates the data for the ListSummary (see page 335).
	GetEnumerator (see page 336)	Gets a System.Collections.IEnumerator for the ListSummary (see page 335).
	GetRssFeed (see page 336)	This method returns the ListSummary (see page 335) RSS feed.
	GetView (see page 336)	Gets the view data using dataset.
	GetViewNames (see page 337)	Gets the view names.
	MoveNext (see page 337)	Move to next index.
	Reset (see page 337)	Resets the index.
	this (see page 337)	Returns the current ContentItem.ContentRow.
	ToString (see page 338)	Overrides the system ToString function.
	Wrap (see page 338)	Wrap the Html toolbar.

ListSummary.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

C#

```
public void CopyTo(Array array, int index);
```

Visual Basic

```
Public Function CopyTo(array As Array, index As Integer) As void
```

ListSummary.Fill Method

Populates the data for the ListSummary (see page 335).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

ListSummary.GetEnumerator Method

Gets a System.Collections.IEnumerator for the ListSummary (see page 335).

C#

```
[Browsable(false)]
public IEnumerator GetEnumerator();
```

Visual Basic

```
<Browsable(false)>
Public Function GetEnumerator() As IEnumerator
```

ListSummary.GetRssFeed Method

This method returns the ListSummary (see page 335) RSS feed.

C#

```
public string GetRssFeed();
```

Visual Basic

```
Public Function GetRssFeed() As string
```

Returns

String

ListSummary.GetView Method

Gets the view data using dataset.

C#

```
public DataSourceView GetView(string viewName);
```

Visual Basic

```
Public Function GetView(viewName As string) As DataSourceView
```

Parameters

Parameters	Description
string viewName	

Returns

DataSourceView

ListSummary.GetViewNames Method

Gets the view names.

C#

```
public ICollection GetViewNames();
```

Visual Basic

```
Public Function GetViewNames() As ICollection
```

Returns

View names

ListSummary.MoveNext Method

Move to next index.

C#

```
public bool MoveNext();
```

Visual Basic

```
Public Function MoveNext() As bool
```

Returns

Next Index

ListSummary.Reset Method

Resets the index.

C#

```
public void Reset();
```

Visual Basic

```
Public Function Reset() As void
```

ListSummary.this Indexer

Returns the current ContentItem.ContentRow.

C#

```
[Browsable(false)]
public Ektron.Cms.Controls.ContentItem.ContentRow this[int i];
```

Visual Basic

```
<Browsable(false)>
Public Sub this(i As Integer)
```

Parameters

Parameters	Description
int i	Index value

Returns

Content (see page 822) Row

ListSummary.ToString Method

Overrides the system ToString function.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Returns

String

ListSummary.Wrap Method

Wrap the Html toolbar.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	

ListSummary Properties

The properties of the ListSummary class are listed here.

Public Properties

	Name	Description
	AddText (see page 339)	Allows overriding the derived controls default Add-menu button text (i.e. 'Add HTML Content (see page 822)' could be changed to 'Add Report')
	CacheInterval (see page 340)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	ContentParameter (see page 340)	Checks the QueryString for this value and replaces the list summary with that content block when specified. Leave blank to always display the list summary.
	ContentType (see page 340)	Determines the type for content to return. For example, Forms. Choices are: • **AllTypes** - displays all of the content types for the given folder. • **Content (see page 822)** - displays a list of content blocks. • **Forms** - forms appear in the list summary. • **Archive_Content** - archived content blocks appear in the list summary. • **Archive_Forms** - archived forms appear in the list summary. • **Assets** - assets, such as offices documents, appear in the list summary. • **Archive_Assets** - archived assets appear in the list summary. • **LibraryItem** - library items appear in the list summary. • **Multimedia** - multimedia items appear in the list summary. • **Archive_Media** -... more (see page 340)
	Count (see page 341)	Indicates the number of elements in your ListSummary (see page 335).

	Current (see page 341)	Gets the current index.
	Direction (see page 341)	Direction for sorting results, "ascending" or "descending".
	EkItems (see page 342)	Returns the array of ContentBase.
	EnablePaging (see page 342)	Enables paging if the total records exceeds MaxResult value.
	FolderDescription (see page 342)	Returns a description of the given folder ID.
	FolderID (see page 343)	The ID of the folder from which to select data. Root is 0.
	FolderName (see page 343)	Returns name of the given folder ID.
	FolderPath (see page 343)	The path of the folder from which to select data. Root starts at \.
	GetHtml (see page 343)	If set to True, retrieves the content (Html body) of every content block in the list summary for display.
	IncludeIcons (see page 344)	Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.
	IsSynchronized (see page 344)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 344)	Gets or Sets the target window for the result.
	MarkupLanguage (see page 345)	Enter the name of a markup file that manages the display of this server control.
	MaxResults (see page 345)	Maximum number of results to be returned. If set to 0 (zero), there is no limit.
	OrderByDirection (see page 345)	Direction (see page 341) for sorting results, "ascending" or "descending".
	OrderKey (see page 346)	The key by which to sort your results.
	Random (see page 346)	If set to true, grabs a single content block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the list summary.
	Recursive (see page 346)	Determines if subfolders of the selected folders should be included in the list.
	SelTaxonomyID (see page 347)	Gets or Sets the TaxonomyID for this Collection (see page 95) Control.
	SuppressWrapperTags (see page 347)	Suppresses the output of span/div tags around the control.
	SyncRoot (see page 347)	Gets an object that can be used to synchronize the ListSummary (see page 335).
	XmlDoc (see page 347)	Returns the Xml object loaded for ListSummary (see page 335).

ListSummary.AddText Property

Allows overriding the derived controls default Add-menu button text (i.e. 'Add HTML Content (see page 822)' could be changed to 'Add Report')

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows overriding the derived controls default Add-menu button text (i.e. 'Add
HTML Content' could be changed to 'Add Report')")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Visible)
]
[Browsable(true)]
public virtual string AddText;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows overriding the derived controls default Add-menu button text (i.e. 'Add
HTML Content' could be changed to 'Add Report')")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Visible)
>
<Browsable(true)>
Public virtual Property AddText() As string
```

ListSummary.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable(), WebBrowsable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable(), WebBrowsable()>
Public virtual Property CacheInterval() As double
```

ListSummary.ContentParameter Property

Checks the QueryString for this value and replaces the list summary with that content block when specified. Leave blank to always display the list summary.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and replaces the list summary with that
content block when specified. Leave blank to always display the list summary.")]
[Personalizable(), WebBrowsable()]
public string ContentParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and replaces the list summary with that
content block when specified. Leave blank to always display the list summary.")>
<Personalizable(), WebBrowsable()>
Public Property ContentParameter() As string
```

ListSummary.ContentType Property

Determines the type for content to return. For example, Forms. Choices are:

- **AllTypes** - displays all of the content types for the given folder.
- **Content (see page 822)** - displays a list of content blocks.
- **Forms** - forms appear in the list summary.
-

Archive_Content - archived content blocks appear in the list summary.

- **Archive_Forms** - archived forms appear in the list summary.
- **Assets** - assets, such as offices documents, appear in the list summary.
- **Archive_Assets** - archived assets appear in the list summary.
- **LibraryItem** - library items appear in the list summary.
- **Multimedia** - multimedia items appear in the list summary.
- **Archive_Media** - archived multimedia items appear in the list summary.
- **NonLibraryContent** - all types of content appear in the list summary except for library items.
- **ImageContent** - images appear in the list summary.
- **DiscussionTopic** - forum topics appear in the list summary.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Detemines the type for content to return. For example, forms.")]
[Personalizable(), WebBrowsable()]
public Ektron.Cms.Controls.CmsWebService.CMSContentType ContentType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Detemines the type for content to return. For example, forms.")>
<Personalizable(), WebBrowsable()>
Public Property ContentType() As Ektron.Cms.Controls.CmsWebService.CMSContentType
```

ListSummary.Count Property

Indicates the number of elements in your ListSummary (see page 335).

C#

```
[Browsable(false)]
public int Count;
```

Visual Basic

```
<Browsable(false)>
Public Property Count() As Integer
```

ListSummary.Current Property

Gets the current index.

C#

```
[Browsable(false)]
public object Current;
```

Visual Basic

```
<Browsable(false)>
Public Property Current() As object
```

ListSummary.Direction Property

Direction for sorting results, "ascending" or "descending".

C#

```
[Obsolete("Use OrderByDirection Property")]
[Browsable(false)]
[Bindable(true)]
```

```
[Category("Ektron")]
[Description("The direction (ascending or descending) that you wish your sorted results to be
displayed.")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Direction"),
WebDescription("Direction")]
public new Ektron.Cms.Controls.CmsWebService.OrderByDirection Direction;
```

Visual Basic

```
<Obsolete("Use OrderByDirection Property")>
<Browsable(false)>
<Bindable(true)>
<Category("Ektron")>
<Description("The direction (ascending or descending) that you wish your sorted results to be
displayed.")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Direction"),
WebDescription("Direction")>
Public new Property Direction() As Ektron.Cms.Controls.CmsWebService.OrderByDirection
```

ListSummary.EkItems Property

Returns the array of ContentBase.

C#

```
[Browsable(false)]
public Ektron.Cms.Common.ContentBase EkItems;
```

Visual Basic

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

ListSummary.EnablePaging Property

Enables paging if the total records exceeds MaxResult value.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enables paging if total records exceeds MaxResult value. Use this property with
non zero value of Maxresults.")]
[Personalizable(), WebBrowsable()]
public bool EnablePaging;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Enables paging if total records exceeds MaxResult value. Use this property with
non zero value of Maxresults.")>
<Personalizable(), WebBrowsable()>
Public Property EnablePaging() As bool
```

ListSummary.FolderDescription Property

Returns a description of the given folder ID.

C#

```
public string FolderDescription;
```

Visual Basic

```
Public Property FolderDescription() As string
```

ListSummary.FolderID Property

The ID of the folder from which to select data. Root is 0.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The ID of the folder from which to select data. Root is 0.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder from which to select data. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property FolderID() As long
```

ListSummary.FolderName Property

Returns name of the given folder ID.

C#

```
public string FolderName;
```

Visual Basic

```
Public Property FolderName() As string
```

ListSummary.FolderPath Property

The path of the folder from which to select data. Root starts at \.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Browsable(false)]
[Description("The path of the folder from which to select data. Root starts at \"\\\".")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Obsolete()]
public string FolderPath;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Browsable(false)>
<Description("The path of the folder from which to select data. Root starts at \"\\\".")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Obsolete()>
Public Property FolderPath() As string
```

ListSummary.GetHtml Property

If set to True, retrieves the content (Html body) of every content block in the list summary for display.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to True, retrieves the content (Html body) of every content block in the
list summary for display.")]
[Personalizable(), WebBrowsable()]
public bool GetHtml;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to True, retrieves the content (Html body) of every content block in the
list summary for display.")>
<Personalizable(), WebBrowsable()>
Public Property GetHtml() As bool
```

ListSummary.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.

C#

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
[Personalizable(), WebBrowsable()]
public bool IncludeIcons;
```

Visual Basic

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")>
<DefaultValue(false)>
<Category("Ektron")>
<Personalizable(), WebBrowsable()>
Public Property IncludeIcons() As bool
```

ListSummary.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

```
[Browsable(false)]
public bool IsSynchronized;
```

Visual Basic

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

ListSummary.LinkTarget Property

Gets or Sets the target window for the result.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
```

```
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

ListSummary.MarkupLanguage Property

Enter the name of a markup file that manages the display of this server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter the name of a markup
file that manages the display of this server control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter the name of a markup
file that manages the display of this server control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

ListSummary.MaxResults Property

Maximum number of results to be returned. If set to 0 (zero), there is no limit.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The maximum number of results to fetch. If set to zero, there is no limit.")]
[Personalizable(), WebBrowsable()]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The maximum number of results to fetch. If set to zero, there is no limit.")>
<Personalizable(), WebBrowsable()>
Public Property MaxResults() As Integer
```

ListSummary.OrderByDirection Property

Direction (see page 341) for sorting results, "ascending" or "descending".

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(EkEnumeration.OrderByDirection.Ascending)]
[Description("Direction for sorting results, 'ascending' or 'descending'.")]
public Ektron.Cms.Common.EkEnumeration.OrderByDirection OrderByDirection;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(EkEnumeration.OrderByDirection.Ascending)>
<Description("Direction for sorting results, 'ascending' or 'descending'.")>
Public Property OrderByDirection() As Ektron.Cms.Common.EkEnumeration.OrderByDirection
```

ListSummary.OrderKey Property

The key by which to sort your results.

C#

```
[Bindable(true)]
[DefaultValue(Ektron.Cms.Controls.CmsWebService.TeasersOrderBy.Title)]
[Description("The key by which to sort your results.")]
[Category("Ektron")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Order"), WebDescription("Order")]
public Ektron.Cms.Controls.CmsWebService.TeasersOrderBy OrderKey;
```

Visual Basic

```
<Bindable(true)>
<DefaultValue(Ektron.Cms.Controls.CmsWebService.TeasersOrderBy.Title)>
<Description("The key by which to sort your results.")>
<Category("Ektron")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Order"), WebDescription("Order")>
Public Property OrderKey() As Ektron.Cms.Controls.CmsWebService.TeasersOrderBy
```

ListSummary.Random Property

If set to true, grabs a single content block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the list summary.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Set this property to True to grab a single, random piece of content from this
list summary.")]
[Personalizable(), WebBrowsable()]
public bool Random;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Set this property to True to grab a single, random piece of content from this
list summary.")>
<Personalizable(), WebBrowsable()>
Public Property Random() As bool
```

ListSummary.Recursive Property

Determines if subfolders of the selected folders should be included in the list.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set recursive to True if you wish to display results in subfolders of your
FolderPath.")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Recursive"), WebDescription("Recursive:
")]
public bool Recursive;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set recursive to True if you wish to display results in subfolders of your
```

```
FolderPath.")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Recursive"), WebDescription("Recursive:
")>
Public Property Recursive() As bool
```

ListSummary.SelTaxonomyID Property

Gets or Sets the TaxonomyID for this Collection (see page 95) Control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Gets or Sets the TaxonomyID for this ListSummary Control.")]
[Personalizable(), WebBrowsable()]
public long SelTaxonomyID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Gets or Sets the TaxonomyID for this ListSummary Control.")>
<Personalizable(), WebBrowsable()>
Public Property SelTaxonomyID() As long
```

ListSummary.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

C#

```
public override bool SuppressWrapperTags;
```

Visual Basic

```
Public override Property SuppressWrapperTags() As bool
```

ListSummary.SyncRoot Property

Gets an object that can be used to synchronize the ListSummary (see page 335).

C#

```
[Browsable(false)]
public object SyncRoot;
```

Visual Basic

```
<Browsable(false)>
Public Property SyncRoot() As object
```

ListSummary.XmlDoc Property

Returns the Xml object loaded for ListSummary (see page 335).

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

Login Class

Summary description for Login.

Class Hierarchy

System.Web.UI.WebControls.WebParts.WebPart → Ektron.Cms.Controls.EkWebControl → Ektron.Cms.Controls.Login

C#

```
public class Login : EkWebControl;
```

Visual Basic

```
Public Class Login
Inherits EkWebControl
```

File

Login.cs

Login Methods

The methods of the Login class are listed here.

Public Methods

	Name	Description
	Fill (see page 348)	Method that can be overridden and used to grab data from the back end for each server control.

Login.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Login Properties

The properties of the Login class are listed here.

Public Properties

	Name	Description
	AutoAddType (see page 348)	Set this property to define auto-added Active Directory (see page 196) users as Members or Authors.
	AutoLogin (see page 349)	Set this property to True if you want to use Auto Login (see page 347) with Active Directory (see page 196).
	OnlyAllowMemberLogin (see page 349)	Set this property to True if you only allow membership users.
	Password (see page 349)	Write Only value for passing in the password.
	PromptLogout (see page 349)	Set this property to False if you do not want to prompt users to verify logout.
	SuppressHelpButton (see page 350)	Set this property to True if you don't want to show the Help button.
	Text (see page 350)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.
	UserName (see page 350)	Write Only value for passing in the UserName.

Login.AutoAddType Property

Set this property to define auto-added Active Directory (see page 196) users as Members or Authors.

C#

```
[Bindable(false), Category("Ektron"), Description("Set this property to define auto-added
```

```
Active Directory users as Members or Authors."),
DefaultValue(Common.EkEnumeration.AutoAddUserTypes.Author)]
public Common.EkEnumeration.AutoAddUserTypes AutoAddType;
```

Visual Basic

```
<Bindable(false), Category("Ektron"), Description("Set this property to define auto-added
Active Directory users as Members or Authors."),
DefaultValue(Common.EkEnumeration.AutoAddUserTypes.Author)>
Public Property AutoAddType() As Common.EkEnumeration.AutoAddUserTypes
```

Login.AutoLogin Property

Set this property to True if you want to use Auto Login (see page 347) with Active Directory (see page 196).

C#

```
[Bindable(false), Category("Ektron"), Description("Set this property to True if you want to
use Auto Login with Active Directory."), DefaultValue(false)]
public bool AutoLogin;
```

Visual Basic

```
<Bindable(false), Category("Ektron"), Description("Set this property to True if you want to
use Auto Login with Active Directory."), DefaultValue(false)>
Public Property AutoLogin() As bool
```

Login.OnlyAllowMemberLogin Property

Set this property to True if you only allow membership users.

C#

```
[Bindable(false), Category("Ektron"), Description("Set this property to True if you only allow
membership users."), DefaultValue(false)]
public bool OnlyAllowMemberLogin;
```

Visual Basic

```
<Bindable(false), Category("Ektron"), Description("Set this property to True if you only allow
membership users."), DefaultValue(false)>
Public Property OnlyAllowMemberLogin() As bool
```

Login.Password Property

Write Only value for passing in the password.

C#

```
public string Password;
```

Visual Basic

```
Public Property Password() As string
```

Login.PromptLogout Property

Set this property to False if you do not want to prompt users to verify logout.

C#

```
[Bindable(false), Category("Ektron"), Description("Set this property to False if you do not
want to prompt users to verify logout."), DefaultValue(true)]
public bool PromptLogout;
```

Visual Basic

```
<Bindable(false), Category("Ektron"), Description("Set this property to False if you do not
want to prompt users to verify logout."), DefaultValue(true)>
Public Property PromptLogout() As bool
```

Login.SuppressHelpButton Property

Set this property to True if you don't want to show the Help button.

C#

```
[Bindable(false), Category("Ektron"), Description("Set this property to True if you don't want
to show the Help button."), DefaultValue(false)]
public bool SuppressHelpButton;
```

Visual Basic

```
<Bindable(false), Category("Ektron"), Description("Set this property to True if you don't want
to show the Help button."), DefaultValue(false)>
Public Property SuppressHelpButton() As bool
```

Login.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

```
public override string Text;
```

Visual Basic

```
Public override Property Text() As string
```

Login.UserName Property

Write Only value for passing in the UserName.

C#

```
public string UserName;
```

Visual Basic

```
Public Property UserName() As string
```

Map Class

Map control

Class Hierarchy

C#

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:Map runat=server></{0}:Map>")]
public class Map : EkWebControl, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:Map runat=server></{0}:Map>")>
Public Class Map
Inherits EkWebControl
Inherits ICallbackEventHandler
```

File

Map.cs

Map Methods

The methods of the Map class are listed here.

Public Methods

	Name	Description
	Fill (see page 351)	
	GetCallbackResult (see page 351)	
	RaiseCallbackEvent (see page 351)	

Map.Fill Method

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Map.GetCallbackResult Method

C#

```
public string GetCallbackResult();
```

Visual Basic

```
Public Function GetCallbackResult() As string
```

Map.RaiseCallbackEvent Method

C#

```
public void RaiseCallbackEvent(string eventArgs);
```

Visual Basic

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

Parameters

Parameters	Description
string eventArgs	

Map Properties

The properties of the Map class are listed here.

Public Properties

	Name	Description
	Address (see page 352)	A default address while loading control. If this value is set, the Latitude (see page 355) and Longitude (see page 356) properties are ignored.
	ContentId (see page 353)	Use this property in conjunction with the EnableSearchResult (see page 354) property to limit the map to a single content item. Here, you identify the content item to be mapped.
	CustomIcon (see page 353)	Custom icon for the category.
	DateSearch (see page 353)	Enable or disable DateTime selector for map data.
	DisplayTextResult (see page 353)	Enable or disable text output for the search results.
	DistanceUnit (see page 354)	Distance measurement units in Miles or Kilometers.

	EnableSearchResult (see page 354)	Determine if the Map (see page 350) server control accesses the CMS search to return results. • **True** = Map (see page 350) server control accesses the CMS search to return results. • **False** = Display a map with only one content item, that is, a single location. Specify this item at the ContentID property. In this case, the map does not find other CMS content whose longitude and latitude are within the map's boundaries. True is the default. Site (see page 725) visitors viewing the map can still use the Search (see page 474) and Directions tabs to get directions to the location. For example, your site features homes for sale, and you... more (see page 354)
	FolderId (see page 354)	The ID of the folder from which to search map data.
	ForceCategory (see page 355)	When set to true, this property causes the map to only show content associated with the taxonomy category defined in the MapCategory (see page 356) property. When set to false, the map shows all content within the maps boundaries. For example, if you have five content blocks that appear on a map and three of them are associated with a taxonomy, set this property to true and the MapCategory (see page 356) property to the ID of the taxonomy. When a user views the map, it will only display the three content items associated with the taxonomy.
	GeoControl (see page 355)	Enable or disable GeoControl which allows a user to perform a search and get directions.
	Latitude (see page 355)	Startup latitude value while loading control.
	Longitude (see page 356)	Startup longitude value while loading control.
	MapCategory (see page 356)	Category to perform search. This category must be created through the taxonomy wizard.
	MapProvider (see page 356)	Provider to connect map server. The supported providers are VirtualEarth and Google.
	MapStyle (see page 356)	Map (see page 350) style accepted values are "Hybrid", "Road" and "Satellite"
	MarkupLanguage (see page 357)	Name of a markup file that manages the display of this server control.
	MinZoomLevel (see page 357)	Search (see page 474) will not be performed beyond this zoom level.
	PageSize (see page 357)	Page Size to populate search result.
	Recursive (see page 357)	Enable or disable recursive search from the specified FolderId (see page 354).
	SearchData (see page 358)	Sets source for the search result.
	SelectedCategoryLogicalType (see page 358)	Logical type operators to search through the selected map categories.
	StartZoomLevel (see page 358)	Startup zoom level. Accepted values between 1 and 19.
	Stylesheet (see page 358)	Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.
	TemplateParamName (see page 359)	Gets or sets a Querystring parameter for the ID of a user.
	TemplateUserProfile (see page 359)	Sets the link for user profile.
	TypeControl (see page 359)	Enables or disables the Google type control and the zoom, direction and type control for Virtual Earth.
	ZoomControl (see page 359)	Enable or Disable zoom control for Google map only.

Map.Address Property

A default address while loading control. If this value is set, the Latitude (see page 355) and Longitude (see page 356) properties are ignored.

C#

```
[Bindable(true), Category("Ektron")]
[Description("A default address while loading control. If this value is set, the Latitude and
Longitude properties are ignored.")]
[Personalizable(), WebBrowsable()]
public string Address;
```

Visual Basic

```
<Bindable(true), Category("Ektron")>
<Description("A default address while loading control. If this value is set, the Latitude and
Longitude properties are ignored.")>
<Personalizable(), WebBrowsable()>
Public Property Address() As string
```

Map.ContentId Property

Use this property in conjunction with the EnableSearchResult (see page 354) property to limit the map to a single content item. Here, you identify the content item to be mapped.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("Gets or Sets the Default ContentID for this Control.")]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long ContentId;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("Gets or Sets the Default ContentID for this Control.")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property ContentId() As long
```

Map.CustomIcon Property

Custom icon for the category.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(false)]
[Description("Custom icon for the category.")]
public bool CustomIcon;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(false)>
<Description("Custom icon for the category.")>
Public Property CustomIcon() As bool
```

Map.DateSearch Property

Enable or disable DateTime selector for map data.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(false)]
[Description("Event time selector for map data.")]
public bool DateSearch;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(false)>
<Description("Event time selector for map data.")>
Public Property DateSearch() As bool
```

Map.DisplayTextResult Property

Enable or disable text output for the search results.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
[Description("Enable or disable text output for the search results.")]
[Personalizable(), WebBrowsable()]
public bool DisplayTextResult;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enable or disable text output for the search results.")>
<Personalizable(), WebBrowsable()>
Public Property DisplayTextResult() As bool
```

Map.DistanceUnit Property

Distance measurement units in Miles or Kilometers.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(Unit.Miles)]
[Description("Distance mesurement units in Miles or Kilometers.")]
public Unit DistanceUnit;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(Unit.Miles)>
<Description("Distance mesurement units in Miles or Kilometers.")>
Public Property DistanceUnit() As Unit
```

Map.EnableSearchResult Property

Determine if the Map (see page 350) server control accesses the CMS search to return results.

- **True** = Map (see page 350) server control accesses the CMS search to return results.
- **False** = Display a map with only one content item, that is, a single location. Specify this item at the ContentID property. In this case, the map does not find other CMS content whose longitude and latitude are within the map's boundaries.

True is the default.

Site (see page 725) visitors viewing the map can still use the Search (see page 474) and Directions tabs to get directions to the location.

For example, your site features homes for sale, and you want a map to show a featured home of the week. To do so, set this property to False, and enter the content block that describes that home in the ContentID field.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
[Description("Enable or disable catalog search.")]
public bool EnableSearchResult;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enable or disable catalog search.")>
Public Property EnableSearchResult() As bool
```

Map.FolderId Property

The ID of the folder from which to search map data.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The ID of the folder from which to search map data.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
```

```
public long FolderId;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder from which to search map data.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderId() As long
```

Map.ForceCategory Property

When set to true, this property causes the map to only show content associated with the taxonomy category defined in the MapCategory (see page 356) property. When set to false, the map shows all content within the maps boundaries.

For example, if you have five content blocks that appear on a map and three of them are associated with a taxonomy, set this property to true and the MapCategory (see page 356) property to the ID of the taxonomy. When a user views the map, it will only display the three content items associated with the taxonomy.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(false)]
[Description("It will force the category filter to MapCategory being applied on startup and on
clearall.")]
public bool ForceCategory;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(false)>
<Description("It will force the category filter to MapCategory being applied on startup and on
clearall.")>
Public Property ForceCategory() As bool
```

Map.GeoControl Property

Enable or disable GeoControl which allows a user to perform a search and get directions.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
[Description("Enable or disable GeoControl which allows a user to perform a search and get
directions.")]
public bool GeoControl;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enable or disable GeoControl which allows a user to perform a search and get
directions.")>
Public Property GeoControl() As bool
```

Map.Latitude Property

Startup latitude value while loading control.

C#

```
[Bindable(true), Category("Ektron")]
[Description("Startup latitude value while loading control.")]
[Personalizable(), WebBrowsable()]
public string Latitude;
```

Visual Basic

```
<Bindable(true), Category("Ektron")>
<Description("Startup latitude value while loading control.")>
<Personalizable(), WebBrowsable()>
Public Property Latitude() As string
```

Map.Longitude Property

Startup longitude value while loading control.

C#

```
[Bindable(true), Category("Ektron")]
[Description("Startup longitude value while loading control.")]
[Personalizable(), WebBrowsable()]
public string Longitude;
```

Visual Basic

```
<Bindable(true), Category("Ektron")>
<Description("Startup longitude value while loading control.")>
<Personalizable(), WebBrowsable()>
Public Property Longitude() As string
```

Map.MapCategory Property

Category to perform search. This category must be created through the taxonomy wizard.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(0)]
[Description("Category to perform search. This category must be created through the taxonomy
wizard.")]
[Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long MapCategory;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(0)>
<Description("Category to perform search. This category must be created through the taxonomy
wizard.")>
<Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property MapCategory() As long
```

Map.MapProvider Property

Provider to connect map server. The supported providers are VirtualEarth and Google.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(Provider.VirtualEarth)]
[Description("Provider to connect map server. The supported providers are VirtualEarth and
Google.")]
[Personalizable(), WebBrowsable()]
public Provider MapProvider;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(Provider.VirtualEarth)>
<Description("Provider to connect map server. The supported providers are VirtualEarth and
Google.")>
<Personalizable(), WebBrowsable()>
Public Property MapProvider() As Provider
```

Map.MapStyle Property

Map (see page 350) style accepted values are "Hybrid", "Road" and "Satellite"

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(Style.Road)]
[Description("Map style accepted values are \"Hybrid\", \"Road\" and \"Satellite\"")]
public Style MapStyle;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(Style.Road)>
<Description("Map style accepted values are \"Hybrid\", \"Road\" and \"Satellite\"")>
Public Property MapStyle() As Style
```

Map.MarkupLanguage Property

Name of a markup file that manages the display of this server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("")]
[Description("Enter the name of a markup file that manages the display of this server
control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("")>
<Description("Enter the name of a markup file that manages the display of this server
control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

Map.MinZoomLevel Property

Search (see page 474) will not be performed beyond this zoom level.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(1)]
[Description("Search will not be performed beyond this zoom level.")]
[Personalizable(), WebBrowsable()]
public int MinZoomLevel;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(1)>
<Description("Search will not be performed beyond this zoom level.")>
<Personalizable(), WebBrowsable()>
Public Property MinZoomLevel() As Integer
```

Map.PageSize Property

Page Size to populate search result.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(10)]
[Description("Page Size to populate search result.")]
[Personalizable(), WebBrowsable()]
public int PageSize;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(10)>
<Description("Page Size to populate search result.")>
<Personalizable(), WebBrowsable()>
Public Property PageSize() As Integer
```

Map.Recursive Property

Enable or disable recursive search from the specified FolderId (see page 354).

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
[Description("Enable or disable recursive search from the specified FolderId.")]
```

```
public bool Recursive;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enable or disable recursive search from the specified FolderId.")>
Public Property Recursive() As bool
```

Map.SearchData Property

Sets source for the search result.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(MapData.Content)]
[Description("Sets source for the search result.")]
[Personalizable(), WebBrowsable()]
public MapData SearchData;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(MapData.Content)>
<Description("Sets source for the search result.")>
<Personalizable(), WebBrowsable()>
Public Property SearchData() As MapData
```

Map.SelectedCategoryLogicalType Property

Logical type operators to search through the selected map categories.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(LogicalType.Or)]
[Description("Logical type operators to search through the selected map categories.")]
public LogicalType SelectedCategoryLogicalType;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(LogicalType.Or)>
<Description("Logical type operators to search through the selected map categories.")>
Public Property SelectedCategoryLogicalType() As LogicalType
```

Map.StartZoomLevel Property

Startup zoom level. Accepted values between 1 and 19.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(13)]
[Description("Startup zoom level. Accepted values between 1 and 19.")]
[Personalizable(), WebBrowsable()]
public int StartZoomLevel;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(13)>
<Description("Startup zoom level. Accepted values between 1 and 19.")>
<Personalizable(), WebBrowsable()>
Public Property StartZoomLevel() As Integer
```

Map.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Category("Ektron")]
[Description("Allows selecting a different CSS file for the map server control.")]
[TypeConverter(typeof(string))]
[DefaultValue("csslib/map.css")]
```

```
public string Stylesheet;
```

Visual Basic

```
<Category("Ektron")>
<Description("Allows selecting a different CSS file for the map server control.")>
<TypeConverter(typeof(string))>
<DefaultValue("csslib/map.css")>
Public Property Stylesheet() As string
```

Map.TemplateParamName Property

Gets or sets a Querystring parameter for the ID of a user.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("Gets or sets a Querystring parameter for the ID of a user.")]
[DefaultValue("id")]
public string TemplateParamName;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("Gets or sets a Querystring parameter for the ID of a user.")>
<DefaultValue("id")>
Public Property TemplateParamName() As string
```

Map.TemplateUserProfile Property

Sets the link for user profile.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("")]
[Description("Sets the link for user profile.")]
public string TemplateUserProfile;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("")>
<Description("Sets the link for user profile.")>
Public Property TemplateUserProfile() As string
```

Map.TypeControl Property

Enables or disables the Google type control and the zoom, direction and type control for Virtual Earth.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
[Description("Enables or disables the Google type control and the zoom, direction and type
control for Virtual Earth.")]
public bool TypeControl;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enables or disables the Google type control and the zoom, direction and type
control for Virtual Earth.")>
Public Property TypeControl() As bool
```

Map.ZoomControl Property

Enable or Disable zoom control for Google map only.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
```

```
[Description("Enable or Disable zoom control for Google map only.")]
public bool ZoomControl;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enable or Disable zoom control for Google map only.")>
Public Property ZoomControl() As bool
```

Membership Class

The Membership server control creates a form on a Web page that allows a user to create or update a profile of themselves in your membership list. The control creates the form based on the standard Ektron user properties and any custom user properties that exist. It also provides a way to have new membership users read and accept the terms and conditions for using Discussion Boards (Forums) when needed.

Class Hierarchy

C#

```
[DefaultProperty("Text"), ToolboxData("<{0}:Membership runat=server></{0}:Membership>")]
public class Membership : EkWebControl, INamingContainer;
```

Visual Basic

```
<DefaultProperty("Text"), ToolboxData("<{0}:Membership runat=server></{0}:Membership>")>
Public Class Membership
Inherits EkWebControl
Inherits INamingContainer
```

File

membership.cs

Membership Methods

The methods of the Membership class are listed here.

Public Methods

	Name	Description
	Fill (see page 360)	Method that can be overridden and used to grab data from the back end for each server control.

Membership.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Membership Properties

The properties of the Membership class are listed here.

Public Properties

	Name	Description
	AllowAvatarUpload (see page 361)	Type of Avatar Upload displayed. Choose from the following ***Active**- Allow users to upload avatar ***(Path To Custom Template)**- Create your own custom template with images and make users to select avatars ***Disable**- Disable avatar upload
	AutoLogin (see page 362)	Set this property to True to automatically log-in after the registration is complete.
	BoardID (see page 362)	Specify the Forum (see page 264) ID to use for the terms and conditions text.
	DisplayMode (see page 362)	Type of membership form displayed. Choose from the following: • **UserRegistration** - lets a site visitor register as a membership user. • **ResetPassword** - lets a membership user reset his password. • **UnsubscribeSecured** - lets a membership user unsubscribe. He needs to enter his username and password. • **UnsubscribeUnsecured** - lets a membership user unsubscribe by entering his membership username only. • **AccountActivate** - let a site visitor activate his membership account by entering the ID number in the account verification email.
	DynamicParameter (see page 363)	To make this control dynamic, select id. When you do, this server control is attached to the user passed as a URL parameter.
	JavascriptEditorHeight (see page 363)	Set the height in pixels for the Signature editor. The default is 300.
	JavascriptEditorWidth (see page 363)	Set the width in pixels for the signature editor. The default is 360.
	RedirectFailedURL (see page 364)	Redirect the user if registering to the site fails.
	RedirectSuccessURL (see page 364)	Redirect the user after successfully registering to the site.
	RegisterButtonImg (see page 364)	Specify the image to use instead of the register button. The image path should be relative to the root.
	RegisterButtonText (see page 364)	Specify the form submit button text.
	ResetButtonImg (see page 365)	Specify the image to use instead of the reset button. The image path should be relative to the root.
	ResetButtonText (see page 365)	Specify the form reset button text.
	ResetPasswordMessageId (see page 365)	To make this control dynamic, select id. When you do, this server control is attached to the user passed as a URL parameter.
	SaveButtonImg (see page 365)	Specify the image to use instead of the save button. The image path should be relative to the root.
	SaveButtonText (see page 366)	Specify the form submit button text when logged in.
	ShowExtended (see page 366)	Include user's custom properties defined in the CMS. The default value is "True".
	ShowTerms (see page 366)	Display the Terms and Conditions text. The default value is "False".
	TaxonomyId (see page 366)	Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.
	Text (see page 367)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.
	UserExistsMessage (see page 367)	User (see page 950) already exists error message.
	UserID (see page 367)	Specify the user ID for which to retrieve information.
	UserSuccessMessage (see page 367)	User (see page 950) successfully registered message.
	UserUpdateSuccessMessage (see page 368)	Message explaining the user successfully update his/her information.

Membership.AllowAvatarUpload Property

Type of Avatar Upload displayed. Choose from the following ***Active**- Allow users to upload avatar ***(Path To Custom**

Template)- Create your own custom template with images and make users to select avatars ***Disable**- Disable avatar upload

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The path to use to render the Avatar Template. (optional)")]
[TypeConverter(typeof(AllowAvatarUploadTypeConverter))]
[DefaultValue("Active")]
public string AllowAvatarUpload;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The path to use to render the Avatar Template. (optional)")>
<TypeConverter(typeof(AllowAvatarUploadTypeConverter))>
<DefaultValue("Active")>
Public Property AllowAvatarUpload() As string
```

Membership.AutoLogin Property

Set this property to True to automatically log-in after the registration is complete.

C#

```
[Bindable(false), Category("Ektron"), Description("Set this property to True to automatically
log-in after the registration is complete."), DefaultValue(true)]
public bool AutoLogin;
```

Visual Basic

```
<Bindable(false), Category("Ektron"), Description("Set this property to True to automatically
log-in after the registration is complete."), DefaultValue(true)>
Public Property AutoLogin() As bool
```

Membership.BoardID Property

Specify the Forum (see page 264) ID to use for the terms and conditions text.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the Forum for which to show terms and conditions.")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BoardID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the Forum for which to show terms and conditions.")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BoardID() As long
```

Membership.DisplayMode Property

Type of membership form displayed. Choose from the following:

- **UserRegistration** - lets a site visitor register as a membership user.
- **ResetPassword** - lets a membership user reset his password.
- **UnsubscribeSecured** - lets a membership user unsubscribe. He needs to enter his username and password.
- **UnsubscribeUnsecured** - lets a membership user unsubscribe by entering his membership username only.
- **AccountActivate** - let a site visitor activate his membership account by entering the ID number in the account verification

email.

C#

```
[Bindable(true), Category("Ektron"), Description("Choose the control's display mode. Each mode
has it own UI and behaves differently. Choices are: UserRegistration, ResetPassword,
UnsubscribeSecured, UnsubscribeUnsecured or AccountActivate."), DefaultValue(0)]
public Mode DisplayMode;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Choose the control's display mode. Each mode
has it own UI and behaves differently. Choices are: UserRegistration, ResetPassword,
UnsubscribeSecured, UnsubscribeUnsecured or AccountActivate."), DefaultValue(0)>
Public Property DisplayMode() As Mode
```

Membership.DynamicParameter Property

To make this control dynamic, select id. When you do, this server control is attached to the user passed as a URL parameter.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("To make this control dynamic, select id. When you do, this server control is
attached to the user passed as a URL parameter.")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("To make this control dynamic, select id. When you do, this server control is
attached to the user passed as a URL parameter.")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

Membership.JavascriptEditorHeight Property

Set the height in pixels for the Signature editor. The default is 300.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(300)]
[Description("The pixel height for the Signature Editor. The default is 300.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
public int JavascriptEditorHeight;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(300)>
<Description("The pixel height for the Signature Editor. The default is 300.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorHeight() As Integer
```

Membership.JavascriptEditorWidth Property

Set the width in pixels for the signature editor. The default is 360.

C#

```
[Bindable(true)]
[Category("Ektron")]
```

```
[DefaultValue(360)]
[Description("The pixel width for the Signature editor. The default is 360.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
public int JavascriptEditorWidth;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(360)>
<Description("The pixel width for the Signature editor. The default is 360.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorWidth() As Integer
```

Membership.RedirectFailedURL Property

Redirect the user if registering to the site fails.

C#

```
[Bindable(true), Category("Ektron"), Description("Redirect the user if registering to the site
fails."), DefaultValue("")]
public string RedirectFailedURL;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Redirect the user if registering to the site
fails."), DefaultValue("")>
Public Property RedirectFailedURL() As string
```

Membership.RedirectSuccessURL Property

Redirect the user after successfully registering to the site.

C#

```
[Bindable(true), Category("Ektron"), Description("Redirect the user after successfully
registering to the site."), DefaultValue("")]
public string RedirectSuccessURL;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Redirect the user after successfully
registering to the site."), DefaultValue("")>
Public Property RedirectSuccessURL() As string
```

Membership.RegisterButtonImg Property

Specify the image to use instead of the register button. The image path should be relative to the root.

C#

```
[Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the
register button. The image path should be relative to the root."), DefaultValue("")]
public string RegisterButtonImg;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the
register button. The image path should be relative to the root."), DefaultValue("")>
Public Property RegisterButtonImg() As string
```

Membership.RegisterButtonText Property

Specify the form submit button text.

C#

```
[Bindable(true), Category("Ektron"), Description("Specify the form submit button text."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_REGBTN)]
```

```
public string RegisterButtonText;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Specify the form submit button text."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_REGBTN)>
Public Property RegisterButtonText() As string
```

Membership.ResetButtonImg Property

Specify the image to use instead of the reset button. The image path should be relative to the root.

C#

```
[Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the
reset button. The image path should be relative to the root."), DefaultValue("")]
public string ResetButtonImg;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the
reset button. The image path should be relative to the root."), DefaultValue("")>
Public Property ResetButtonImg() As string
```

Membership.ResetButtonText Property

Specify the form reset button text.

C#

```
[Bindable(true), Category("Ektron"), Description("Specify the form reset button text."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_RESTBTN)]
public string ResetButtonText;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Specify the form reset button text."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_RESTBTN)>
Public Property ResetButtonText() As string
```

Membership.ResetPasswordMessageId Property

To make this control dynamic, select id. When you do, this server control is attached to the user passed as a URL parameter.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The id of the message to send out for the reset password email.")]
[DefaultValue(0)]
public long ResetPasswordMessageId;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The id of the message to send out for the reset password email.")>
<DefaultValue(0)>
Public Property ResetPasswordMessageId() As long
```

Membership.SaveButtonImg Property

Specify the image to use instead of the save button. The image path should be relative to the root.

C#

```
[Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the save
button. The image path should be relative to the root."), DefaultValue("")]
```

```
public string SaveButtonImg;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the save
button. The image path should be relative to the root."), DefaultValue("")>
Public Property SaveButtonImg() As string
```

Membership.SaveButtonText Property

Specify the form submit button text when logged in.

C#

```
[Bindable(true), Category("Ektron"), Description("Specify the form submit button text when
logged in."), TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_SAVBTN)]
public string SaveButtonText;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Specify the form submit button text when
logged in."), TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_SAVBTN)>
Public Property SaveButtonText() As string
```

Membership.ShowExtended Property

Include user's custom properties defined in the CMS. The default value is "True".

C#

```
[Bindable(true), Category("Ektron"), Description("Include user's custom properties defined in
the CMS. The default value is \"True\"."), DefaultValue(true)]
public bool ShowExtended;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Include user's custom properties defined in
the CMS. The default value is \"True\"."), DefaultValue(true)>
Public Property ShowExtended() As bool
```

Membership.ShowTerms Property

Display the Terms and Conditions text. The default value is "False".

C#

```
[Bindable(true), Category("Ektron"), Description("Display the Terms and Conditions text. The
default value is \"False\"."), DefaultValue(false)]
public bool ShowTerms;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Display the Terms and Conditions text. The
default value is \"False\"."), DefaultValue(false)>
Public Property ShowTerms() As bool
```

Membership.TaxonomyId Property

Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)]
[Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")]
[Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long TaxonomyId;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)>
<Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")>
<Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property TaxonomyId() As long
```

Membership.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

```
public override string Text;
```

Visual Basic

```
Public override Property Text() As string
```

Membership.UserExistsMessage Property

User (see page 950) already exists error message.

C#

```
[Bindable(true), Category("Ektron"), Description("User already exists error message."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_EXISTS)]
public string UserExistsMessage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("User already exists error message."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_EXISTS)>
Public Property UserExistsMessage() As string
```

Membership.UserID Property

Specify the user ID for which to retrieve information.

C#

```
[Bindable(false), Browsable(false), Description("Specify the user ID for which to retrieve
information."), DefaultValue("0")]
public long UserID;
```

Visual Basic

```
<Bindable(false), Browsable(false), Description("Specify the user ID for which to retrieve
information."), DefaultValue("0")>
Public Property UserID() As long
```

Membership.UserSuccessMessage Property

User (see page 950) successfully registered message.

C#

```
[Bindable(true), Category("Ektron"), Description("User successfully registered message."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_SUCCEED)]
public string UserSuccessMessage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("User successfully registered message."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
```

```
DefaultValue(EkWebControl.RES_MEMBR_SUCCEED)>
Public Property UserSuccessMessage() As string
```

Membership.UserUpdateSuccessMessage Property

Message explaining the user successfully update his/her information.

C#

```
[Bindable(true), Category("Ektron"), Description("User successfully update his/her
information."), TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_UPDATE)]
public string UserUpdateSuccessMessage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("User successfully update his/her
information."), TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_UPDATE)>
Public Property UserUpdateSuccessMessage() As string
```

Menu Class

The Menu server control calls a menu to be displayed on a page. Using the Menu server control, you can manipulate a menu by using the DisplayXslt (see page 370) property.

Class Hierarchy

C#

```
[DefaultProperty("DefaultMenuID"), ToolboxData("<{0}:Menu runat=server></{0}:Menu>"),
PersistChildren(true)]
public class Menu : EkXsltBase;
```

Visual Basic

```
<DefaultProperty("DefaultMenuID"), ToolboxData("<{0}:Menu runat=server></{0}:Menu>"),
PersistChildren(true)>
Public Class Menu
Inherits EkXsltBase
```

File

Menu.cs

Menu Methods

The methods of the Menu class are listed here.

Public Methods

	Name	Description
	Fill (see page 368)	Calls up the database and updates the contents of the Menu (see page 368) if the data is out of sync with the specified parameters.
	Wrap (see page 369)	Method that can overridden and be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Menu.Fill Method

Calls up the database and updates the contents of the Menu (see page 368) if the data is out of sync with the specified parameters.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Menu.Wrap Method

Method that can overridden and be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

Html string wrapped with a toolbar.

Menu Properties

The properties of the Menu class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 369)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DefaultMenuID (see page 370)	Gets or Sets the Default MenuID for this Control.
	DisplayXslt (see page 370)	The Xslt to use to render the Menu (see page 368) (optional)
	DynamicParameter (see page 370)	The parameter to query for a Dynamic Menu (see page 368).
	Stylesheet (see page 371)	Specify the style sheet relative to site root. Leave blank for default.
	SuppressAddEdit (see page 371)	Setting this property to True caused the menu to hide the Add and Edit buttons for the menu. It is defaulted to false.

Menu.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
```

```
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
Public virtual Property CacheInterval() As double
```

Menu.DefaultMenuID Property

Gets or Sets the Default MenuID for this Control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Editor(typeof(MenuUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Description("Gets or Sets the Default MenuID for this Control.")]
public long DefaultMenuID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Editor(typeof(MenuUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("Gets or Sets the Default MenuID for this Control.")>
Public Property DefaultMenuID() As long
```

Menu.DisplayXslt Property

The Xslt to use to render the Menu (see page 368) (optional)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the Menu")]
[TypeConverter(typeof(XsltMenuTypeConverter))]
[DefaultValue("SampleMenu")]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the Menu")>
<TypeConverter(typeof(XsltMenuTypeConverter))>
<DefaultValue("SampleMenu")>
Public override Property DisplayXslt() As string
```

Menu.DynamicParameter Property

The parameter to query for a Dynamic Menu (see page 368).

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[Description("The parameter to query for a Dynamic Menu.")]
[TypeConverter(typeof(DynamicMenuTypeConverter))]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("None - Use Default")>
<Description("The parameter to query for a Dynamic Menu.")>
<TypeConverter(typeof(DynamicMenuTypeConverter))>
Public Property DynamicParameter() As string
```

Menu.Stylesheet Property

Specify the style sheet relative to site root. Leave blank for default.

C#

```
[Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(StringConverter))]
public virtual string Stylesheet;
```

Visual Basic

```
<Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(StringConverter))>
Public virtual Property Stylesheet() As string
```

Menu.SuppressAddEdit Property

Setting this property to True caused the menu to hide the Add and Edit buttons for the menu. It is defaulted to false.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Setting this property to true causes the menu to hide the Add and Edit buttons
for the menu. It is defaulted to false.")]
[DefaultValue(false)]
public virtual bool SuppressAddEdit;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Setting this property to true causes the menu to hide the Add and Edit buttons
for the menu. It is defaulted to false.")>
<DefaultValue(false)>
Public virtual Property SuppressAddEdit() As bool
```

MessageBoard Class

Summary description for Message board

Class Hierarchy

C#

```
[DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:MessageBoard
runat=server></{0}:MessageBoard>")]
public class MessageBoard : EkXsltWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:MessageBoard
runat=server></{0}:MessageBoard>")>
Public Class MessageBoard
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

File

MessageBoard.cs

MessageBoard Methods

The methods of the MessageBoard class are listed here.

Public Methods

	Name	Description
	CustomSpamMethod (see page 372)	This method exposes the custom spam delegate that is invoked when a use selects the custom option.
	Fill (see page 372)	Populates the data for the MessageBoard (see page 371).
	RaiseCallbackEvent (see page 372)	
	ToString (see page 372)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 373)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

MessageBoard.CustomSpamMethod Method

This method exposes the custom spam delegate that is invoked when a use selects the custom option.

C#

```
public void CustomSpamMethod(CustomSpamHandler MethodAddress);
```

Visual Basic

```
Public Function CustomSpamMethod(MethodAddress As CustomSpamHandler) As void
```

Parameters

Parameters	Description
methodPointer	The method that implements the custom spam functionality.

MessageBoard.Fill Method

Populates the data for the MessageBoard (see page 371).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

MessageBoard.RaiseCallbackEvent Method

C#

```
public void RaiseCallbackEvent(string eventArgs);
```

Visual Basic

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

Parameters

Parameters	Description
string eventArgs	string

MessageBoard.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

MessageBoard.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

MessageBoard Properties

The properties of the MessageBoard class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 374)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	CommentBoxLocation (see page 374)	This property will place the Add Comments TextArea on top or bottom.
	DefaultObjectID (see page 374)	The default object ID for this control to use when there is no matching dynamic parameter value passed.
	DisplayXslt (see page 375)	Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.
	DynamicObjectParameter (see page 375)	Gets or sets the Querystring parameter to read a object ID dynamically. To use the default object ID, leave blank.
	EnablePaging (see page 375)	Enables paging if the total records exceeds MaxResult value. This property, in conjunction with the MaxResults (see page 376) property, lets site visitors view an unlimited number of document items while controlling the amount of screen space. To accomplish this, the display is limited to the number set in the MaxResults (see page 376) property.If you set this property to true, and the number of items exceeds the MaxResults (see page 376) number, navigation aids appear below the last item.
	MarkupLanguage (see page 376)	Identify the template markup file that controls the display of the server control. For example, mymarkup.ekml. If the *.ekml file is located in the same folder as the Web form containing the server control, just enter its name. Otherwise, the path to the file can be relative or absolute.
	MaxResults (see page 376)	Maximum number of results to be returned.
	Moderate (see page 377)	This property indicates if the messages are to be moderated by the administrator of the site.
	ObjectType (see page 377)	The type of object to which the message board is assigned. Choices are: • **Content (see page 822)** - the message board is assigned to a content item. • **User (see page 950)** - the message board is assigned to an individual. • **CommunityGroup (see page 750)** - the message board is assigned to a community group.
	ProfileParamName (see page 377)	The name of the parameter to pass to the profile template.
	ShowMaxCharacter (see page 378)	The maximum number of characters to display in the message. If the number of characters in the message surpasses the number set in this property, a "more..." link appears. Clicking this link displays the full text of the message.
	SpamControlType (see page 378)	This property prevents spamming of the message board.

	SpamTimeSpan (see page 378)	This property sets the time for spam control in seconds.
	Stylesheet (see page 379)	Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet.
	TemplateUserProfile (see page 379)	The URL to launch a new window with, when a member clicks the display-name shown in the messages' browse users screen (used to get more information on a potential message recipient). Optional.
	XmlDoc (see page 379)	Returns the Xslt object for the collection.

MessageBoard.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

MessageBoard.CommentBoxLocation Property

This property will place the Add Comments TextArea on top or bottom.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("This property will place the Add Comments TextArea on top or bottom.")]
[DefaultValue(EkEnumeration.MessageBoardCommentBoxLocation.Bottom)]
public EkEnumeration.MessageBoardCommentBoxLocation CommentBoxLocation;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("This property will place the Add Comments TextArea on top or bottom.")>
<DefaultValue(EkEnumeration.MessageBoardCommentBoxLocation.Bottom)>
Public Property CommentBoxLocation() As EkEnumeration.MessageBoardCommentBoxLocation
```

MessageBoard.DefaultObjectID Property

The default object ID for this control to use when there is no matching dynamic parameter value passed.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
```

```
[Description("The default object ID for this control to use when there is no matching dynamic
parameter value passed.")]
public long DefaultObjectID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The default object ID for this control to use when there is no matching dynamic
parameter value passed.")>
Public Property DefaultObjectID() As long
```

MessageBoard.DisplayXslt Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltMessageBoardDataTypeConverter))]
[DefaultValue("Builtin")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltMessageBoardDataTypeConverter))>
<DefaultValue("Builtin")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

MessageBoard.DynamicObjectParameter Property

Gets or sets the Querystring parameter to read a object ID dynamically. To use the default object ID, leave blank.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Gets or sets the Querystring parameter to read a object ID dynamically. To use
the default object ID, leave blank.")]
[Personalizable()]
public string DynamicObjectParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Gets or sets the Querystring parameter to read a object ID dynamically. To use
the default object ID, leave blank.")>
<Personalizable()>
Public Property DynamicObjectParameter() As string
```

MessageBoard.EnablePaging Property

Enables paging if the total records exceeds MaxResult value. This property, in conjunction with the MaxResults (see page 376) property, lets site visitors view an unlimited number of document items while controlling the amount of screen space. To accomplish this, the display is limited to the number set in the MaxResults (see page 376) property.If you set this property to

true, and the number of items exceeds the MaxResults (see page 376) number, navigation aids appear below the last item.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Enables paging if total records exceeds MaxResult value. This property, in
conjunction with the MaxResults property, lets site visitors view an unlimited number of
document items while controlling the amount of screen space. To accomplish this, the display
is limited to the number set in the MaxResults property. If you set this property to true, and
the number of items exceeds the MaxResults number, navigation aids appear below the last
item.")]
[Personalizable()]
public bool EnablePaging;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Enables paging if total records exceeds MaxResult value. This property, in
conjunction with the MaxResults property, lets site visitors view an unlimited number of
document items while controlling the amount of screen space. To accomplish this, the display
is limited to the number set in the MaxResults property. If you set this property to true, and
the number of items exceeds the MaxResults number, navigation aids appear below the last
item.")>
<Personalizable()>
Public Property EnablePaging() As bool
```

MessageBoard.MarkupLanguage Property

Identify the template markup file that controls the display of the server control. For example, mymarkup.ekml. If the *.ekml file is located in the same folder as the Web form containing the server control, just enter its name. Otherwise, the path to the file can be relative or absolute.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter the name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter the name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

MessageBoard.MaxResults Property

Maximum number of results to be returned.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of results to fetch. If set to zero, there is no limit.")]
[Personalizable()]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("The maximum number of results to fetch. If set to zero, there is no limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

MessageBoard.Moderate Property

This property indicates if the messages are to be moderated by the administrator of the site.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("This property indicates if the messages are to be moderated by the administrator
of the site.")]
[DefaultValue(false)]
public bool Moderate;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("This property indicates if the messages are to be moderated by the administrator
of the site.")>
<DefaultValue(false)>
Public Property Moderate() As bool
```

MessageBoard.ObjectType Property

The type of object to which the message board is assigned. Choices are:

- **Content (see page 822)** - the message board is assigned to a content item.
- **User (see page 950)** - the message board is assigned to an individual.
- **CommunityGroup (see page 750)** - the message board is assigned to a community group.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Select the type of object with which this control is associated. Choices are:
Content, User and CommunityGroup.")]
[DefaultValue(EkEnumeration.MessageBoardObjectType.Content)]
public EkEnumeration.MessageBoardObjectType ObjectType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Select the type of object with which this control is associated. Choices are:
Content, User and CommunityGroup.")>
<DefaultValue(EkEnumeration.MessageBoardObjectType.Content)>
Public Property ObjectType() As EkEnumeration.MessageBoardObjectType
```

MessageBoard.ProfileParamName Property

The name of the parameter to pass to the profile template.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The parameter name to pass on the querystring to the profile and location pages,
which will be set to the ID of the user that was clicked on.")]
[DefaultValue("id")]
public string ProfileParamName;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The parameter name to pass on the querystring to the profile and location pages,
which will be set to the ID of the user that was clicked on.")>
<DefaultValue("id")>
```

```
Public Property ProfileParamName() As string
```

MessageBoard.ShowMaxCharacter Property

The maximum number of characters to display in the message. If the number of characters in the message surpasses the number set in this property, a “more...” link appears. Clicking this link displays the full text of the message.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The maximum number of characters to display in the message. If the number of
characters in the message surpasses the number set in this property, a ?more...? link appears.
Clicking this link displays the full text of the message.")]
[Personalizable()]
public int ShowMaxCharacter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The maximum number of characters to display in the message. If the number of
characters in the message surpasses the number set in this property, a ?more...? link appears.
Clicking this link displays the full text of the message.")>
<Personalizable()>
Public Property ShowMaxCharacter() As Integer
```

MessageBoard.SpamControlType Property

This property prevents spamming of the message board.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("This property prevents spamming of the message board.")]
[DefaultValue(EkEnumeration.MessageBoardSpamControlType.None)]
public EkEnumeration.MessageBoardSpamControlType SpamControlType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("This property prevents spamming of the message board.")>
<DefaultValue(EkEnumeration.MessageBoardSpamControlType.None)>
Public Property SpamControlType() As EkEnumeration.MessageBoardSpamControlType
```

MessageBoard.SpamTimeSpan Property

This property sets the time for spam control in seconds.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("This property sets the time for spam control in seconds.")]
[DefaultValue(0)]
public int SpamTimeSpan;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("This property sets the time for spam control in seconds.")>
<DefaultValue(0)>
Public Property SpamTimeSpan() As Integer
```

MessageBoard.Stylesheet Property

Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control. Select either 'Use Default', 'None', or enter a
path to a custom Css file. The location can be relative or absolute.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control. Select either 'Use Default', 'None', or enter a
path to a custom Css file. The location can be relative or absolute.")>
<Personalizable()>
Public Property Stylesheet() As string
```

MessageBoard.TemplateUserProfile Property

The URL to launch a new window with, when a member clicks the display-name shown in the messages' browse users screen (used to get more information on a potential message recipient). Optional.

C#

```
[Category("Ektron")]
[Description("The URL to launch a new window with, when a member clicks the display-name shown
in the messages' browse users screen (used to get more information on a potential message
recipient). Optional.")]
[TypeConverter(typeof(string))]
[DefaultValue("")]
public string TemplateUserProfile;
```

Visual Basic

```
<Category("Ektron")>
<Description("The URL to launch a new window with, when a member clicks the display-name shown
in the messages' browse users screen (used to get more information on a potential message
recipient). Optional.")>
<TypeConverter(typeof(string))>
<DefaultValue("")>
Public Property TemplateUserProfile() As string
```

MessageBoard.XmlDoc Property

Returns the Xslt object for the collection.

C#

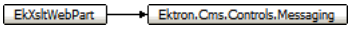
```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

Messaging Class

Class Hierarchy



C#

```
[ToolboxData("<{0}:Messaging runat=server></{0}:Messaging>")]
public class Messaging : EkXsltWebPart;
```

Visual Basic

```
<ToolboxData("<{0}:Messaging runat=server></{0}:Messaging>")>
Public Class Messaging
Inherits EkXsltWebPart
```

File

Messaging.cs

Messaging Methods

The methods of the Messaging class are listed here.

Public Methods

	Name	Description
	Fill (see page 380)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 380)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 380)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Messaging.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Messaging.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Messaging.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

Messaging Properties

The properties of the Messaging class are listed here.

Public Properties

	Name	Description
	DisplayXslt (see page 381)	Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. (optional)
	EnablePreSearch (see page 382)	Determines if the message recipient user search should render with a completed search for all when opened.
	FriendsOnly (see page 382)	When this property is set to True, users can only send messages to people on their colleagues list. Default is True. • **True** - send messages to colleagues only. • **False** - send messages to all CMS400.NET users.
	GroupParamName (see page 382)	Enter the QueryString parameter used to pass the ID of a group. For example, if the QueryString is: `?gid=1`, then enter “gid” in this property. You can only pass one group. This property is typically used when you want to pre-populate the **To:** field of a message.
	JavascriptEditorHeight (see page 383)	Set the height in pixels for the editor. The default is 400.
	JavascriptEditorHTMLMode (see page 383)	Set to True to allow users to edit their posts and replies in HTML. The default is False. See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Allowing HTML Editing in the Javascript Editor"
	JavascriptEditorToolbar (see page 383)	Decide which buttons are available in the Javascript Editor. For additional information on the Javascript Toolbar Editor, See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Adding and Removing Toolbar Buttons From the Javascript Editor"
	JavascriptEditorWidth (see page 384)	Set the width in pixels for the editor. The default is 625.
	ProfileParamName (see page 384)	This property identifies the QueryString parameter used to pass the ID of a user to a profile or location template when a user name is clicked in the Browse User (see page 950) screen. The default is ID. The parameter defined in this property is appended to the QueryString of the path defined in the ProfileUrl (see page 385) property. The default is ID. **Important:** This parameter needs to match the parameter set in the UserProfile (see page 520) server control’s DynamicParameter property on the page defined in the ProfileUrl (see page 385) property.
	ProfileUrl (see page 385)	The URL of the profile page template. This page is opened in a new window when a user clicks a display name on the Browse Users (see page 1076) screen. This allows a user to get additional profile information about the potential message recipient. (Optional)
	RecipientParamName (see page 385)	Enter the QueryString parameter used to pass the ID of a user. For example, if the QueryString is: `?uid=1`, then enter “uid” in this property. You can pass multiple users by repeating this. For example: `?uid=1&uid=20&uid=12&uid=18` This property is typically used when you want to pre-populate the **To:** field of a message.
	ShowSearchOptions (see page 385)	This property is obsolete and no longer has any affect.
	Stylesheet (see page 386)	Specify the path to a style sheet for use with the server control. The location can be relative or absolute. Leave blank to use the default style sheet.
	UserSearchDisplayXslt (see page 386)	The XSLT used to display the user search control inside the Messaging (see page 380) server control.
	XmlDoc (see page 386)	Returns the Xslt object for the collection.

Messaging.DisplayXslt Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. (optional)

C#

```
[Category("Ektron")]
```

```
[Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute. (optional)")]
[TypeConverter(typeof(XsltMessagingTypeConverter))]
[DefaultValue("Basic Messaging")]
public override string DisplayXslt;
```

Visual Basic

```
<Category("Ektron")>
<Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute. (optional)")>
<TypeConverter(typeof(XsltMessagingTypeConverter))>
<DefaultValue("Basic Messaging")>
Public override Property DisplayXslt() As string
```

Messaging.EnablePreSearch Property

Determines if the message recipient user search should render with a completed search for all when opened.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set this to true to have all users (or colleagues) listed when the search is
opened, otherwise setting false will prevent this.")]
public bool EnablePreSearch;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set this to true to have all users (or colleagues) listed when the search is
opened, otherwise setting false will prevent this.")>
Public Property EnablePreSearch() As bool
```

Messaging.FriendsOnly Property

When this property is set to True, users can only send messages to people on their colleagues list. Default is True.

- **True** - send messages to colleagues only.
- **False** - send messages to all CMS400.NET users.

C#

```
[Category("Ektron")]
[DefaultValue(true)]
[Description("When set to false, includes all CMS users when True only friends will be
available as message recipients. True is default.")]
public bool FriendsOnly;
```

Visual Basic

```
<Category("Ektron")>
<DefaultValue(true)>
<Description("When set to false, includes all CMS users when True only friends will be
available as message recipients. True is default.")>
Public Property FriendsOnly() As bool
```

Messaging.GroupParamName Property

Enter the QueryString parameter used to pass the ID of a group. For example, if the QueryString is: `?gid=1`, then enter “gid” in this property. You can only pass one group. This property is typically used when you want to pre-populate the **To:** field of a message.

C#

```
[Category("Ektron")]
```

```
[TypeConverter(typeof(string))]
[Description("The parameter name to pass on the query string, that identifies the group for
the message (used to pre-populate the to=field)")]
[DefaultValue("gid")]
public string GroupParamName;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The parameter name to pass on the query string, that identifies the group for
the message (used to pre-populate the to=field)")>
<DefaultValue("gid")>
Public Property GroupParamName() As string
```

Messaging.JavascriptEditorHeight Property

Set the height in pixels for the editor. The default is 400.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(400)]
[Description("The pixel height for the Javascript Editor.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
public int JavascriptEditorHeight;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(400)>
<Description("The pixel height for the Javascript Editor.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorHeight() As Integer
```

Messaging.JavascriptEditorHTMLMode Property

Set to True to allow users to edit their posts and replies in HTML. The default is False. See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Allowing HTML Editing in the Javascript Editor"

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Determines if HTML mode is allowed for the Javascript Editor.")]
[Obsolete("JavaScript Editor has been deprecated. Instead control is using CMS400WebEditor.")]
public bool JavascriptEditorHTMLMode;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Determines if HTML mode is allowed for the Javascript Editor.")>
<Obsolete("JavaScript Editor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorHTMLMode() As bool
```

Messaging.JavascriptEditorToolbar Property

Decide which buttons are available in the Javascript Editor. For additional information on the Javascript Toolbar Editor, See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Adding and Removing Toolbar Buttons From the Javascript Editor"

C#

```
[Bindable(true)]
```

```
[Category("Ektron")]
[DefaultValue("ParagraphMenu,FontFacesMenu,FontSizesMenu,FontForeColorsMenu|Bold,Italic,Underli
ne,Strikethrough;Superscript,Subscript,RemoveFormat|JustifyLeft,JustifyCenter,JustifyRight,Just
ifyFull;BulletedList,NumberedList,Indent,Outdent;CreateLink,Unlink,InsertRule|Cut,Copy,Paste;Un
do,Redo,Print")]
[Description("The toolbar configuration for the Javascript Editor.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
public string JavascriptEditorToolbar;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("ParagraphMenu,FontFacesMenu,FontSizesMenu,FontForeColorsMenu|Bold,Italic,Underli
ne,Strikethrough;Superscript,Subscript,RemoveFormat|JustifyLeft,JustifyCenter,JustifyRight,Just
ifyFull;BulletedList,NumberedList,Indent,Outdent;CreateLink,Unlink,InsertRule|Cut,Copy,Paste;Un
do,Redo,Print")>
<Description("The toolbar configuration for the Javascript Editor.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorToolbar() As string
```

Messaging.JavascriptEditorWidth Property

Set the width in pixels for the editor. The default is 625.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(625)]
[Description("The pixel width for the Javascript Editor.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
public int JavascriptEditorWidth;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(625)>
<Description("The pixel width for the Javascript Editor.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorWidth() As Integer
```

Messaging.ProfileParamName Property

This property identifies the QueryString parameter used to pass the ID of a user to a profile or location template when a user name is clicked in the Browse User (see page 950) screen. The default is ID. The parameter defined in this property is appended to the QueryString of the path defined in the ProfileUrl (see page 385) property. The default is ID. **Important:** This parameter needs to match the parameter set in the UserProfile (see page 520) server control's DynamicParameter property on the page defined in the ProfileUrl (see page 385) property.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("This property identifies the QueryString parameter used to pass the ID of a user
to a profile page or location template when a user name is clicked in the Browse User screen.
The default is ID. The parameter defined in this property is appended to the QueryString of
the path defined in the ProfileUrl property. The default is ID.")]
[DefaultValue("id")]
public string ProfileParamName;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("This property identifies the QueryString parameter used to pass the ID of a user
to a profile page or location template when a user name is clicked in the Browse User screen.
```

```
The default is ID. The parameter defined in this property is appended to the QueryString of
the path defined in the ProfileUrl property. The default is ID.")>
<DefaultValue("id")>
Public Property ProfileParamName() As string
```

Messaging.ProfileUrl Property

The URL of the profile page template. This page is opened in a new window when a user clicks a display name on the Browse Users (see page 1076) screen. This allows a user to get additional profile information about the potential message recipient. (Optional)

C#

```
[Category("Ektron")]
[Description("The URL of the profile page template. This page is opened when a user clicks a
display name in the Browse Users screen. This allows a user to get additional profile
information about the potential message recipient. (Optional)")]
[TypeConverter(typeof(string))]
[DefaultValue("")]
public string ProfileUrl;
```

Visual Basic

```
<Category("Ektron")>
<Description("The URL of the profile page template. This page is opened when a user clicks a
display name in the Browse Users screen. This allows a user to get additional profile
information about the potential message recipient. (Optional)")>
<TypeConverter(typeof(string))>
<DefaultValue("")>
Public Property ProfileUrl() As string
```

Messaging.RecipientParamName Property

Enter the QueryString parameter used to pass the ID of a user. For example, if the QueryString is: `?uid=1`, then enter “uid” in this property. You can pass multiple users by repeating this. For example: `?uid=1&uid=20&uid=12&uid=18` This property is typically used when you want to pre-populate the **To:** field of a message.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The parameter name to pass on the querystring, that identifies the recipient of
the message (used to prepopulate the to=field)")]
[DefaultValue("uid")]
public string RecipientParamName;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The parameter name to pass on the querystring, that identifies the recipient of
the message (used to prepopulate the to=field)")>
<DefaultValue("uid")>
Public Property RecipientParamName() As string
```

Messaging.ShowSearchOptions Property

This property is obsolete and no longer has any affect.

C#

```
[Category("Ektron")]
[DefaultValue(false)]
[Description("This property is obsolete and no longer has any affect")]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool ShowSearchOptions;
```

Visual Basic

```
<Category("Ektron")>
<DefaultValue(false)>
<Description("This property is obsolete and no longer has any affect")>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property ShowSearchOptions() As bool
```

Messaging.Stylesheet Property

Specify the path to a style sheet for use with the server control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with the server control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[TypeConverter(typeof(string))]
[DefaultValue("Community.css")]
public string Stylesheet;
```

Visual Basic

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with the server control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<TypeConverter(typeof(string))>
<DefaultValue("Community.css")>
Public Property Stylesheet() As string
```

Messaging.UserSearchDisplayXslt Property

The XSLT used to display the user search control inside the Messaging (see page 380) server control.

C#

```
[Category("Ektron")]
[Description("The XSLT used to display the user search control inside the Messaging server
control.")]
[TypeConverter(typeof(String))]
[DefaultValue("MsgUserSelect.xsl")]
public string UserSearchDisplayXslt;
```

Visual Basic

```
<Category("Ektron")>
<Description("The XSLT used to display the user search control inside the Messaging server
control.")>
<TypeConverter(typeof(String))>
<DefaultValue("MsgUserSelect.xsl")>
Public Property UserSearchDisplayXslt() As string
```

Messaging.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

MetaData Class

The Metadata server control lets you add the metadata from content blocks to a Web page. This lets developers add metadata

quickly without having to type it in. You can add metadata from a single content block, multiple content blocks, or by dynamically passing a content ID from a URL.

Class Hierarchy

System.Web.UI.WebControls.WebParts.WebPart → Ektron.Cms.Controls.EkWebControl → Ektron.Cms.Controls.MetaData

C#

```
public class MetaData : Ektron.Cms.Controls.EkWebControl;
```

Visual Basic

```
Public Class MetaData
Inherits Ektron.Cms.Controls.EkWebControl
```

File

MetaData.cs

MetaData Methods

The methods of the MetaData class are listed here.

Public Methods

	Name	Description
	Fill (see page 387)	Method that can be overridden and used to grab data from the back end for each server control.

MetaData.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

MetaData Properties

The properties of the MetaData class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 388)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DefaultContentID (see page 388)	The content block ID from which the server control gets the metadata. If you want to add metadata from several content blocks, set this property to 0 (zero) and use the DefaultItemList (see page 389) property to identify the content blocks.
	DefaultItemList (see page 389)	A comma delimited list of content block IDs from which to get metadata. DefaultContentID (see page 388) must be set to 0 (zero) so you can add IDs to the list.
	DynamicParameter (see page 389)	Gets or sets the QueryString parameter to read a content ID dynamically. Set to "Use Default" if you wish to always display the default ContentID (static.)

	GenerateDublinCore (see page 389)	When enabled, this property automatically creates seven of the Simple Dublin Core (see page 1047) metadata fields from standard CMS400.NET system properties. The default is false. • **True** = generate Simple Dublin Core (see page 1047) metadata fields • **False** = do not generate Simple Dublin Core (see page 1047) metadata fields The seven fields and how they are associated with the CMS400.NET properties is explained in the CMS400.NET Developer Manual section "MetaData (see page 386) Server Control" > "Using the Simple Dublin Core (see page 1047) Metadata (see page 714) Standard"

MetaData.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable(), WebBrowsable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable(), WebBrowsable()>
Public virtual Property CacheInterval() As double
```

MetaData.DefaultContentID Property

The content block ID from which the server control gets the metadata. If you want to add metadata from several content blocks, set this property to 0 (zero) and use the DefaultItemList (see page 389) property to identify the content blocks.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the Default ContentID for this Control.")]
public long DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the Default ContentID for this Control.")>
Public Property DefaultContentID() As long
```

MetaData.DefaultItemList Property

A comma delimited list of content block IDs from which to get metadata. DefaultContentID (see page 388) must be set to 0 (zero) so you can add IDs to the list.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("A comma delimited list of content block IDs from which to get metadata.
DefaultContentID must be set to 0 (zero) so you can add IDs to the list.")]
public string DefaultItemList;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("A comma delimited list of content block IDs from which to get metadata.
DefaultContentID must be set to 0 (zero) so you can add IDs to the list.")>
Public Property DefaultItemList() As string
```

MetaData.DynamicParameter Property

Gets or sets the QueryString parameter to read a content ID dynamically. Set to "Use Default" if you wish to always display the default ContentID (static.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Content ID dynamically - Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Content ID dynamically - Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

MetaData.GenerateDublinCore Property

When enabled, this property automatically creates seven of the Simple Dublin Core (see page 1047) metadata fields from standard CMS400.NET system properties. The default is false.

- **True** = generate Simple Dublin Core (see page 1047) metadata fields
- **False** = do not generate Simple Dublin Core (see page 1047) metadata fields

The seven fields and how they are associated with the CMS400.NET properties is explained in the CMS400.NET Developer Manual section "MetaData (see page 386) Server Control" > "Using the Simple Dublin Core (see page 1047) Metadata (see page 714) Standard"

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Append Dublin Core Metadata")]
public bool GenerateDublinCore;
```

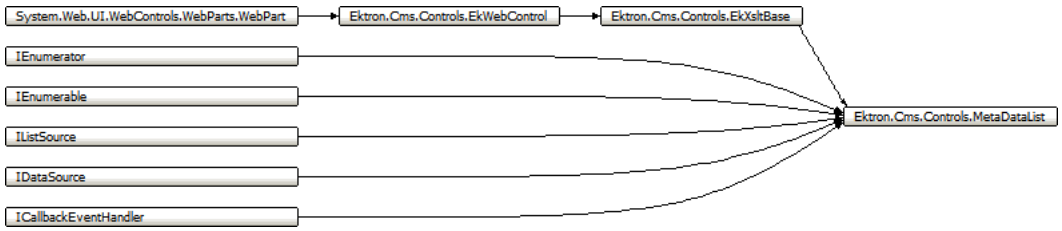
Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Append Dublin Core Metadata")>
Public Property GenerateDublinCore() As bool
```

MetaDataList Class

Ektron ContentByMeta class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.

Class Hierarchy



C#

```
[DefaultProperty("KeyWordName"), PersistChildren(true)]
public class MetaDataList : EkXsltBase, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("KeyWordName"), PersistChildren(true)>
Public Class MetaDataList
Inherits EkXsltBase
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
Inherits ICallbackEventHandler
```

File

MetaDataList.cs

MetaDataList Methods

The methods of the MetaDataList class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 391)	Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.
	Fill (see page 391)	Calls up the database and updates the items in the control if the data is out of sync with the specified parameters.
	GetEnumerator (see page 391)	Gets a System.Collections.IEnumerator for the collection.
	MoveNext (see page 391)	Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.
	Reset (see page 391)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 391)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.
	Wrap (see page 392)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

MetaDataList.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

C#

```
public void CopyTo(Array array, int i);
```

Visual Basic

```
Public Function CopyTo(array As Array, i As Integer) As void
```

MetaDataList.Fill Method

Calls up the database and updates the items in the control if the data is out of sync with the specified parameters.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

MetaDataList.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

```
public IEnumerator GetEnumerator();
```

Visual Basic

```
Public Function GetEnumerator() As IEnumerator
```

MetaDataList.MoveNext Method

Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.

C#

```
public bool MoveNext();
```

Visual Basic

```
Public Function MoveNext() As bool
```

Returns

Boolean indicating if the move was within the range of the array.

MetaDataList.Reset Method

Brings the index of the collection back to zero. Does not clear the collection.

C#

```
public void Reset();
```

Visual Basic

```
Public Function Reset() As void
```

MetaDataList.this Indexer

Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.

C#

```
public Ektron.Cms.Controls.ContentItem.ContentRow this[int i];
```

Visual Basic

```
Public Sub this(i As Integer)
```

MetaDataList.Wrap Method

Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	HTML string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

MetaDataList Properties

The properties of the MetaDataList class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 393)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	ContentType (see page 394)	Select a type of content for this control. Choices are: • all types • archived content • archived forms • content • forms
	Count (see page 394)	Indicates the number of elements in your Collection (see page 95).
	Current (see page 395)	Gets the current object in the collection.
	EkItems (see page 395)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 95). Holds more information than the exposed strongly typed this[i] index operator.
	EnablePaging (see page 395)	This property, in conjunction with the `MaxNumber (see page 399)` property, lets site visitors view an unlimited number of content items while controlling the amount of screen space. To accomplish this, the content display is limited to the number set in the `MaxNumber (see page 399)` property. If you set this property to **True**, and the number of content items exceeds the `MaxNumber (see page 399)` number, navigation aids appear below the last item.
	ExactPhrase (see page 395)	Gets or Sets the boolean value of ExactPhrase. Determines whether the KeyWordValue (see page 398) needs to match the phrase exactly. For example, if "site" is the KeyWordValue (see page 398) and the title of a content block is "Welcome (see page 1) to the site" and ExactPhrase is set to **True**, you would not see the content block in the metadata list. This is because "site" does not equal "Welcome (see page 1) to the site". • **True** = match the exact phrase • **False** = does not need to match exact phrase

	FolderID (see page 396)	The folder ID from which content is retrieved. At the `Recursive (see page 400)` property, you determine if content in this folder's subfolders is also retrieved.
	GetHtml (see page 396)	If set to true, retrieves the content (Html body) of every content block in the MetadataList for display.
	IncludeIcons (see page 397)	Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with "ecmSummary" and "ecmTeaser" or when the [$imageIcon] variable is used in an EkML file.
	IsSynchronized (see page 397)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	KeyWordName (see page 397)	KeyWordName represents a metadata definition, that is, the container for the KeyWordValues. Examples of a KeyWordName are **Keywords** and **Title**.
	KeyWordValue (see page 398)	Enter the values associated with the `KeyWordName (see page 397)`. Only content whose metadata (defined at the KeyWordName (see page 397) property) matches this value appears on the metadata list.
	KeyWordValueMatchAll (see page 398)	This property is only used if you enter more than one keyword value. If you do, and only want content to appear on the metadata list if all values entered at the KeyWordValue (see page 398) field match its metadata values, enter true. If metadata can appear on the list as long as *any value* defined at the KeyWordValue (see page 398) field matches the selected metadata value for a content item, enter **false**. **Example:** KeyWordValue (see page 398) for **Title** (assigned for this server control): home; page; company. Metadata (see page 714) values for a content item's Title metadata definition field: software; ektron; company. If `KeyWordValueMatchAll = true`, content... more (see page 398)
	KeyWordValueSeparator (see page 398)	Enter the character used to separate the list of keyword values. An example is a semicolon(**;**).
	LinkTarget (see page 399)	Gets or Sets the target window for the result. Choices are: • **_blank** - this target causes the link to always be loaded in a new blank window. This window is not named. • **_self** - this target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target. • **_parent** - this target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** - this target makes the link load in the full... more (see page 399)
	MarkupLanguage (see page 399)	Enter name of a markup file that manages the display of this server control.
	MaxNumber (see page 399)	Determine the maximum number of content items to return. If set to zero then all items are returned.
	OrderBy (see page 400)	The order of the list to be returned. Order by: • **Title** - The title of the content block • **ID** - The content block ID number • **Date Created** - The date the content block was created • **Date Modified** - The date the content block was last modified • **LastEditorLname** - The last editor's last name • **LastEditorFname** - The last editor's first name
	Recursive (see page 400)	Whether to search sub-folders of the identified root folder. The starting folder is identified in the `FolderID (see page 396)` property.
	SortOrder (see page 401)	Gets or sets the sort order. Choose whether to sort by Ascending or Descending.
	SuppressWrapperTags (see page 401)	Suppresses the output of span/div tags around the control.
	SyncRoot (see page 401)	Gets an object that can be used to synchronize the collection.
	XmlDoc (see page 401)	Returns the XSLT object for the MetaDataList (see page 390) control.

MetaDataList.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable(), WebBrowsable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable(), WebBrowsable()>
Public virtual Property CacheInterval() As double
```

MetaDataList.ContentType Property

Select a type of content for this control. Choices are:

- all types
- archived content
- archived forms
- content
- forms

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Select a type of content for this control. Choices are: all types, archived
content, archived forms, content or forms.")]
[Personalizable(), WebBrowsable()]
public Ektron.Cms.Controls.CmsWebService.CMSContentType ContentType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Select a type of content for this control. Choices are: all types, archived
content, archived forms, content or forms.")>
<Personalizable(), WebBrowsable()>
Public Property ContentType() As Ektron.Cms.Controls.CmsWebService.CMSContentType
```

MetaDataList.Count Property

Indicates the number of elements in your Collection (see page 95).

C#

```
[Category("Data")]
[Description("Indicates the number of elements in your Collection.")]
public int Count;
```

Visual Basic

```
<Category("Data")>
<Description("Indicates the number of elements in your Collection.")>
```

```
Public Property Count() As Integer
```

MetaDataList.Current Property

Gets the current object in the collection.

C#

```
[Browsable(false)]
public object Current;
```

Visual Basic

```
<Browsable(false)>
Public Property Current() As object
```

MetaDataList.EkItems Property

Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 95). Holds more information than the exposed strongly typed this[i] index operator.

C#

```
[Browsable(false)]
public Ektron.Cms.Common.ContentBase EkItems;
```

Visual Basic

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

MetaDataList.EnablePaging Property

This property, in conjunction with the `MaxNumber` (see page 399) property, lets site visitors view an unlimited number of content items while controlling the amount of screen space. To accomplish this, the content display is limited to the number set in the `MaxNumber` (see page 399) property. If you set this property to **True**, and the number of content items exceeds the `MaxNumber` (see page 399) number, navigation aids appear below the last item.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enables paging if total records exceeds MaxResult value. Use this property with
a non zero value in the MaxNumbers properties.")]
public bool EnablePaging;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Enables paging if total records exceeds MaxResult value. Use this property with
a non zero value in the MaxNumbers properties.")>
Public Property EnablePaging() As bool
```

MetaDataList.ExactPhrase Property

Gets or Sets the boolean value of ExactPhrase. Determines whether the KeyWordValue (see page 398) needs to match the phrase exactly.

For example, if “site” is the KeyWordValue (see page 398) and the title of a content block is “Welcome (see page 1) to the site” and ExactPhrase is set to **True**, you would not see the content block in the metadata list. This is because “site” does not equal “Welcome (see page 1) to the site”.

- **True** = match the exact phrase
- **False** = does not need to match exact phrase

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("True")]
[Description("Gets or Sets the boolean value of ExactPhrase. Determines whether the
KeyWordValue needs to match the phrase exactly. For example, if \"site\" is the KeyWordValue
and the title of a content block is \"Welcome to the site\" and ExactPhrase is set to
\"True\", you would not see the content block in the metadata list. This is because \"site\"
does not equal \"Welcome to the site\".")]
public bool ExactPhrase;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("True")>
<Description("Gets or Sets the boolean value of ExactPhrase. Determines whether the
KeyWordValue needs to match the phrase exactly. For example, if \"site\" is the KeyWordValue
and the title of a content block is \"Welcome to the site\" and ExactPhrase is set to
\"True\", you would not see the content block in the metadata list. This is because \"site\"
does not equal \"Welcome to the site\".")>
Public Property ExactPhrase() As bool
```

MetaDataList.FolderID Property

The folder ID from which content is retrieved. At the `Recursive` (see page 400) property, you determine if content in this folder's subfolders is also retrieved.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[Description("The folder ID from which content is retrieved. At the \"Recursive property\",
you determine if content in this folder?s subfolders is also retrieved.")]
[Editor(typeof(FolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("None - Use Default")>
<Description("The folder ID from which content is retrieved. At the \"Recursive property\",
you determine if content in this folder?s subfolders is also retrieved.")>
<Editor(typeof(FolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property FolderID() As long
```

MetaDataList.GetHtml Property

If set to true, retrieves the content (Html body) of every content block in the MetadataList for display.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, retrieves the content (Html body) of every content block in the
MetadataList for display.")]
public bool GetHtml;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
```

```
<Description("If set to true, retrieves the content (Html body) of every content block in the
MetadataList for display.")>
Public Property GetHtml() As bool
```

MetaDataList.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with "ecmSummary" and "ecmTeaser" or when the [$imageIcon] variable is used in an EkML file.

C#

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
public bool IncludeIcons;
```

Visual Basic

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")>
<DefaultValue(false)>
<Category("Ektron")>
Public Property IncludeIcons() As bool
```

MetaDataList.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

```
[Browsable(false)]
public bool IsSynchronized;
```

Visual Basic

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

MetaDataList.KeyWordName Property

KeyWordName represents a metadata definition, that is, the container for the KeyWordValues. Examples of a KeyWordName are **Keywords** and **Title**.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(" -None -")]
[Description("KeyWordName represents a metadata definition, that is, the container for the
KeyWordValues. Examples of a KeyWordName are \"Keywords\" and \"Title\".")]
[Editor(typeof(MetaDataUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public string KeyWordName;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(" -None -")>
<Description("KeyWordName represents a metadata definition, that is, the container for the
KeyWordValues. Examples of a KeyWordName are \"Keywords\" and \"Title\".")>
<Editor(typeof(MetaDataUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property KeyWordName() As string
```

MetaDataList.KeyWordValue Property

Enter the values associated with the `KeyWordName` (see page 397). Only content whose metadata (defined at the KeyWordName (see page 397) property) matches this value appears on the metadata list.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[Description("Enter the values associated with the \"KeyWordName\". Only content whose
metadata (defined at the KeyWordName property) matches this value appears on the metadata
list.")]
[Personalizable(), WebBrowsable()]
public string KeyWordValue;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("None - Use Default")>
<Description("Enter the values associated with the \"KeyWordName\". Only content whose
metadata (defined at the KeyWordName property) matches this value appears on the metadata
list.")>
<Personalizable(), WebBrowsable()>
Public Property KeyWordValue() As string
```

MetaDataList.KeyWordValueMatchAll Property

This property is only used if you enter more than one keyword value.

If you do, and only want content to appear on the metadata list if all values entered at the KeyWordValue (see page 398) field match its metadata values, enter true.

If metadata can appear on the list as long as *any value* defined at the KeyWordValue (see page 398) field matches the selected metadata value for a content item, enter **false**.

Example:

KeyWordValue (see page 398) for **Title** (assigned for this server control): home; page; company.

Metadata (see page 714) values for a content item's Title metadata definition field: software; ektron; company.

If `KeyWordValueMatchAll = true`, content does not appear on metadata list because some items do not match.

If `KeyWordValueMatchAll = false`, content item appears on metadata list because one item (company) matches.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Match all values specified in KeyWordValue separator by KeyWordValueSeparator.")]
public bool KeyWordValueMatchAll;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Match all values specified in KeyWordValue separator by KeyWordValueSeparator.")>
Public Property KeyWordValueMatchAll() As bool
```

MetaDataList.KeyWordValueSeparator Property

Enter the character used to separate the list of keyword values. An example is a semicolon(**;**).

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(";")]
[Description("Enter the character used to separate the list of keyword values. An example is a
semicolon(;).")]
[Personalizable(), WebBrowsable()]
public string KeyWordValueSeparator;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(";")>
<Description("Enter the character used to separate the list of keyword values. An example is a
semicolon(;).")>
<Personalizable(), WebBrowsable()>
Public Property KeyWordValueSeparator() As string
```

MetaDataList.LinkTarget Property

Gets or Sets the target window for the result. Choices are:

- **_blank** - this target causes the link to always be loaded in a new blank window. This window is not named.
- **_self** - this target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target.
- **_parent** - this target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.
- **_top** - this target makes the link load in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

MetaDataList.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

MetaDataList.MaxNumber Property

Determine the maximum number of content items to return. If set to zero then all items are returned.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Determine the maximum number of content items to return. If set to zero then all
items are returned.")]
public int MaxNumber;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Determine the maximum number of content items to return. If set to zero then all
items are returned.")>
Public Property MaxNumber() As Integer
```

MetaDataList.OrderBy Property

The order of the list to be returned. Order by:

- **Title** - The title of the content block
- **ID** - The content block ID number
- **Date Created** - The date the content block was created
- **Date Modified** - The date the content block was last modified
- **LastEditorLname** - The last editor's last name
- **LastEditorFname** - The last editor's first name

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("This value determines how to order the return list. Choices are: Title, ID. Date
Created, Date Modified, LastEditorLname or LastEditorFname.")]
[Personalizable(), WebBrowsable()]
public Ektron.Cms.Controls.CmsWebService.ContentOrderBy OrderBy;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("This value determines how to order the return list. Choices are: Title, ID. Date
Created, Date Modified, LastEditorLname or LastEditorFname.")>
<Personalizable(), WebBrowsable()>
Public Property OrderBy() As Ektron.Cms.Controls.CmsWebService.ContentOrderBy
```

MetaDataList.Recursive Property

Whether to search sub-folders of the identified root folder. The starting folder is identified in the `FolderID` (see page 396) property.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Whether to search sub-folders of the identified root folder. The starting folder
is identified in the \"FolderID\" property.")]
public bool Recursive;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
```

```
<DefaultValue(false)>
<Description("Whether to search sub-folders of the identified root folder. The starting folder
is identified in the \"FolderID\" property.")>
Public Property Recursive() As bool
```

MetaDataList.SortOrder Property

Gets or sets the sort order. Choose whether to sort by Ascending or Descending.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("This value determines how to order the return list, Ascending or Descending.")]
[Personalizable(), WebBrowsable()]
public Ektron.Cms.Controls.CmsWebService.OrderByDirection SortOrder;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("This value determines how to order the return list, Ascending or Descending.")>
<Personalizable(), WebBrowsable()>
Public Property SortOrder() As Ektron.Cms.Controls.CmsWebService.OrderByDirection
```

MetaDataList.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

C#

```
public override bool SuppressWrapperTags;
```

Visual Basic

```
Public override Property SuppressWrapperTags() As bool
```

MetaDataList.SyncRoot Property

Gets an object that can be used to synchronize the collection.

C#

```
[Browsable(false)]
public object SyncRoot;
```

Visual Basic

```
<Browsable(false)>
Public Property SyncRoot() As object
```

MetaDataList.XmlDoc Property

Returns the XSLT object for the MetaDataList (see page 390) control.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

MicroMessaging Class

Summary description for MicroMessage (see page 1070)

Class Hierarchy

C#

```
[DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:MicroMessaging
runat=server></{0}:MicroMessaging>")]
public class MicroMessaging : EkXsltWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:MicroMessaging
runat=server></{0}:MicroMessaging>")>
Public Class MicroMessaging
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

File

MicroMessaging.cs

MicroMessaging Methods

The methods of the MicroMessaging class are listed here.

Public Methods

	Name	Description
	CustomSpamMethod (see page 402)	This method exposes the custom spam delegate that is invoked when a use selects the custom option.
	Fill (see page 402)	Populates the data for the MicroMessaging (see page 401) List.
	RaiseCallbackEvent (see page 402)	
	ToString (see page 403)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 403)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

MicroMessaging.CustomSpamMethod Method

This method exposes the custom spam delegate that is invoked when a use selects the custom option.

C#

```
public void CustomSpamMethod(CustomSpamControlHandler MethodAddress);
```

Visual Basic

```
Public Function CustomSpamMethod(MethodAddress As CustomSpamControlHandler) As void
```

Parameters

Parameters	Description
methodPointer	The method that implements the custom spam functionality.

MicroMessaging.Fill Method

Populates the data for the MicroMessaging (see page 401) List.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

MicroMessaging.RaiseCallbackEvent Method

C#

```
public void RaiseCallbackEvent(string eventArgs);
```

Visual Basic

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

Parameters

Parameters	Description
string eventArgs	string

MicroMessaging.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

MicroMessaging.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

MicroMessaging Properties

The properties of the MicroMessaging class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 404)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DefaultObjectID (see page 404)	The default object ID for this control to use when there is no matching dynamic parameter value passed.
	DisplayXslt (see page 405)	Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.
	DynamicObjectParameter (see page 405)	Gets or sets the Querystring parameter to read a object ID dynamically. To use the default object ID, leave blank.
	EnablePaging (see page 405)	Enables paging if the total records exceeds MaxResult value. This property, in conjunction with the MaxResults (see page 406) property, lets site visitors view an unlimited number of document items while controlling the amount of screen space. To accomplish this, the display is limited to the number set in the MaxResults (see page 406) property.If you set this property to true, and the number of items exceeds the MaxResults (see page 406) number, navigation aids appear below the last item.
	MaxResults (see page 406)	Maximum number of results to be returned.
	MaxSearchResults (see page 406)	Maximum number of search results to be returned.

	Mode (see page 407)	The mode in which the micro messaging server control is going to be used on the template. • **User (see page 950)** - the micro message control displays the logged in user's status or statuses. • **TimeLine** - the micro message control displays all public and colleague's statuses for the logged in user.
	ProfileParamName (see page 407)	The name of the parameter to pass to the profile template.
	SpamControlType (see page 407)	This property prevents spamming of the micro messaging control.
	SpamTimeSpan (see page 408)	This property sets the time for spam control in seconds.
	Stylesheet (see page 408)	Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet.
	TemplateMessage (see page 408)	The URL to launch a new window with only the message, when a member clicks the time lapse for any message(used to get URL for a single message). Optional.
	TemplateUserProfile (see page 409)	The URL to launch a new window with, when a member clicks the display-name shown in the messages' browse users screen (used to get more information on a potential message recipient). Optional.
	XmlDoc (see page 409)	Returns the Xslt object for the collection.

MicroMessaging.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

MicroMessaging.DefaultObjectID Property

The default object ID for this control to use when there is no matching dynamic parameter value passed.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The default object ID for this control to use when there is no matching dynamic
parameter value passed.")]
public long DefaultObjectID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The default object ID for this control to use when there is no matching dynamic
parameter value passed.")>
Public Property DefaultObjectID() As long
```

MicroMessaging.DisplayXslt Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltMicroMessagingTypeConverter))]
[DefaultValue("Builtin")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltMicroMessagingTypeConverter))>
<DefaultValue("Builtin")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

MicroMessaging.DynamicObjectParameter Property

Gets or sets the Querystring parameter to read a object ID dynamically. To use the default object ID, leave blank.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Gets or sets the Querystring parameter to read a object ID dynamically. To use
the default object ID, leave blank.")]
[Personalizable()]
public string DynamicObjectParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Gets or sets the Querystring parameter to read a object ID dynamically. To use
the default object ID, leave blank.")>
<Personalizable()>
Public Property DynamicObjectParameter() As string
```

MicroMessaging.EnablePaging Property

Enables paging if the total records exceeds MaxResult value. This property, in conjunction with the MaxResults (see page 406) property, lets site visitors view an unlimited number of document items while controlling the amount of screen space. To accomplish this, the display is limited to the number set in the MaxResults (see page 406) property.If you set this property to true, and the number of items exceeds the MaxResults (see page 406) number, navigation aids appear below the last item.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Enables paging if total records exceeds MaxResult value. This property, in
conjunction with the MaxResults property, lets site visitors view an unlimited number of
document items while controlling the amount of screen space. To accomplish this, the display
is limited to the number set in the MaxResults property. If you set this property to true, and
the number of items exceeds the MaxResults number, navigation aids appear below the last
item.")]
[Personalizable()]
public bool EnablePaging;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Enables paging if total records exceeds MaxResult value. This property, in
conjunction with the MaxResults property, lets site visitors view an unlimited number of
document items while controlling the amount of screen space. To accomplish this, the display
is limited to the number set in the MaxResults property. If you set this property to true, and
the number of items exceeds the MaxResults number, navigation aids appear below the last
item.")>
<Personalizable()>
Public Property EnablePaging() As bool
```

MicroMessaging.MaxResults Property

Maximum number of results to be returned.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of results to fetch. If set to zero, there is no limit.")]
[Personalizable()]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("The maximum number of results to fetch. If set to zero, there is no limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

MicroMessaging.MaxSearchResults Property

Maximum number of search results to be returned.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of search results to fetch. If set to zero, there is no
limit.")]
[Personalizable()]
public int MaxSearchResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
```

```
<Description("The maximum number of search results to fetch. If set to zero, there is no
limit.")>
<Personalizable()>
Public Property MaxSearchResults() As Integer
```

MicroMessaging.Mode Property

The mode in which the micro messaging server control is going to be used on the template.

- **User (see page 950)** - the micro message control displays the logged in user's status or statuses.
- **TimeLine** - the micro message control displays all public and colleague's statuses for the logged in user.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Select the mode in which the control is to be displayed. Choices are: Status,and
TimeLine.")]
[DefaultValue(EkEnumeration.MicroMessageMode.User)]
public EkEnumeration.MicroMessageMode Mode;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Select the mode in which the control is to be displayed. Choices are: Status,and
TimeLine.")>
<DefaultValue(EkEnumeration.MicroMessageMode.User)>
Public Property Mode() As EkEnumeration.MicroMessageMode
```

MicroMessaging.ProfileParamName Property

The name of the parameter to pass to the profile template.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The parameter name to pass on the querystring to the profile and location pages,
which will be set to the ID of the user that was clicked on.")]
[DefaultValue("id")]
public string ProfileParamName;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The parameter name to pass on the querystring to the profile and location pages,
which will be set to the ID of the user that was clicked on.")>
<DefaultValue("id")>
Public Property ProfileParamName() As string
```

MicroMessaging.SpamControlType Property

This property prevents spamming of the micro messaging control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("This property prevents spamming of the micro messaging control.")]
[DefaultValue(EkEnumeration.MicroMessageSpamControlType.None)]
public EkEnumeration.MicroMessageSpamControlType SpamControlType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("This property prevents spamming of the micro messaging control.")>
```

```
<DefaultValue(EkEnumeration.MicroMessageSpamControlType.None)>
Public Property SpamControlType() As EkEnumeration.MicroMessageSpamControlType
```

MicroMessaging.SpamTimeSpan Property

This property sets the time for spam control in seconds.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("This property sets the time for spam control in seconds.")]
[DefaultValue(0)]
public int SpamTimeSpan;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("This property sets the time for spam control in seconds.")>
<DefaultValue(0)>
Public Property SpamTimeSpan() As Integer
```

MicroMessaging.Stylesheet Property

Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control. Select either 'Use Default', 'None', or enter a
path to a custom Css file. The location can be relative or absolute.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control. Select either 'Use Default', 'None', or enter a
path to a custom Css file. The location can be relative or absolute.")>
<Personalizable()>
Public Property Stylesheet() As string
```

MicroMessaging.TemplateMessage Property

The URL to launch a new window with only the message, when a member clicks the time lapse for any message(used to get URL for a single message). Optional.

C#

```
[Category("Ektron")]
[Description("The URL to launch a new window with only the message, when a member clicks the
time lapse for any message(used to get URL for a single message). Optional.")]
[TypeConverter(typeof(string))]
[DefaultValue("")]
public string TemplateMessage;
```

Visual Basic

```
<Category("Ektron")>
<Description("The URL to launch a new window with only the message, when a member clicks the
time lapse for any message(used to get URL for a single message). Optional.")>
```

```
<TypeConverter(typeof(string))>
<DefaultValue("")>
Public Property TemplateMessage() As string
```

MicroMessaging.TemplateUserProfile Property

The URL to launch a new window with, when a member clicks the display-name shown in the messages' browse users screen (used to get more information on a potential message recipient). Optional.

C#

```
[Category("Ektron")]
[Description("The URL to launch a new window with, when a member clicks the display-name shown
in the messages' browse users screen (used to get more information on a potential message
recipient). Optional.")]
[TypeConverter(typeof(string))]
[DefaultValue("")]
public string TemplateUserProfile;
```

Visual Basic

```
<Category("Ektron")>
<Description("The URL to launch a new window with, when a member clicks the display-name shown
in the messages' browse users screen (used to get more information on a potential message
recipient). Optional.")>
<TypeConverter(typeof(string))>
<DefaultValue("")>
Public Property TemplateUserProfile() As string
```

MicroMessaging.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

MyAccount Class

Class Hierarchy

C#

```
[DefaultProperty("DefaultMyAccountID"), ToolboxData("<{0}:MyAccount
runat=server></{0}:MyAccount>")]
public class MyAccount : ECommerceControl, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultMyAccountID"), ToolboxData("<{0}:MyAccount
runat=server></{0}:MyAccount>")>
Public Class MyAccount
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

File

MyAccount.cs

Description

This is class Ektron.Cms.Controls.MyAccount.

MyAccount Methods

The methods of the MyAccount class are listed here.

Public Methods

	Name	Description
	Fill (see page 410)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 410)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 410)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

MyAccount.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

MyAccount.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

MyAccount.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

MyAccount Properties

The properties of the MyAccount class are listed here.

Public Properties

	Name	Description
	CustomPropertyID (see page 411)	An ID or comma separated list of IDs for a Custom User (see page 950) Property (or properties) to expose in the UI. These properties are used to define information about a user beyond the standard Ektron CMS400.NET user properties; such as Username, First Name, Password and email Address.

	DefaultCountryId (see page 411)	The ID of the default country that appears in the Billing Address and Shipping Address. Set this ID to the country from which the majority of your customers will be making purchases. If the site visitor is from a different country, he can it change when editing the Billing or Shipping Address. **Note:** If a site visitor enters information in the Billing Address area, his Billing Country seeds the Shipping Country when adding a new Shipping Address.
	DisplayXslt (see page 412)	Specify an external XSLT file to render the control. By default, the control uses MyAccount.xsl. This file is located in
	EkItem (see page 412)	Provides access to the CustomerData (see page 1478) object.
	LoadingImage (see page 412)	Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**.
	RedirectUrl (see page 413)	Enter the path of a template that contains the Login (see page 347) server control. When a site visitor arrives at this control and is not logged in, the control checks this property for a path to template containing a Login (see page 347) server control. If one exists, the site visitor is sent to that template. Once he's logged in, he's returned to the template containing the MyAccount (see page 409) server control. If no path exists, the control checks the following key the web.config file for a path.
	Stylesheet (see page 413)	Specify the path to a style sheet for use with this server control. The location can be relative or absolute. Leave blank to use the default style sheet.
	UseMsAjax (see page 413)	When true, uses Microsoft Ajax.

MyAccount.CustomPropertyID Property

An ID or comma separated list of IDs for a Custom User (see page 950) Property (or properties) to expose in the UI. These properties are used to define information about a user beyond the standard Ektron CMS400.NET user properties; such as Username, First Name, Password and email Address.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("An ID or comma separated list of IDs for a Custom User Property (or properties)
to expose in the UI. These properties are used to define information about a user beyond the
standard Ektron CMS400.NET user properties; such as Username, First Name, Password and email
Address.")]
[DefaultValue("")]
public string CustomPropertyID;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("An ID or comma separated list of IDs for a Custom User Property (or properties)
to expose in the UI. These properties are used to define information about a user beyond the
standard Ektron CMS400.NET user properties; such as Username, First Name, Password and email
Address.")>
<DefaultValue("")>
Public Property CustomPropertyID() As string
```

MyAccount.DefaultCountryId Property

The ID of the default country that appears in the Billing Address and Shipping Address. Set this ID to the country from which the majority of your customers will be making purchases. If the site visitor is from a different country, he can it change when editing the Billing or Shipping Address. **Note:** If a site visitor enters information in the Billing Address area, his Billing Country seeds the Shipping Country when adding a new Shipping Address.

C#

```
[Category("Ektron")]
[Description("The numeric ID of the country to use by default, which is then used to obtain
regions/states when country is unknown (for USA, DefaultCountryId=438).")]
public int DefaultCountryId;
```

Visual Basic

```
<Category("Ektron")>
<Description("The numeric ID of the country to use by default, which is then used to obtain
regions/states when country is unknown (for USA, DefaultCountryId=438).")>
Public Property DefaultCountryId() As Integer
```

MyAccount.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses MyAccount.xsl. This file is located in

```
[Site Root]\Workarea\Xslt\Commerce
```

Important: If you want to edit this file, it is strongly recommended that you make a copy, change its name and move it outside of the Workarea folder. If you make changes to this file and do not move it out of the Workarea, this file and the changes will be lost when you upgrade.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses MyAccount.xsl. This file is located in [Site Root]\\Workarea\\Xslt\\Commerce. Important:
If you want to edit this file, it is strongly recommended that you make a copy, change its
name and move it outside of the Workarea folder. If you make changes to this file and do not
move it out of the Workarea, this file and the changes will be lost when you upgrade.")]
[TypeConverter(typeof(XsltMyAccountTypeConverter))]
[DefaultValue("MyAccount.xsl")]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses MyAccount.xsl. This file is located in [Site Root]\\Workarea\\Xslt\\Commerce. Important:
If you want to edit this file, it is strongly recommended that you make a copy, change its
name and move it outside of the Workarea folder. If you make changes to this file and do not
move it out of the Workarea, this file and the changes will be lost when you upgrade.")>
<TypeConverter(typeof(XsltMyAccountTypeConverter))>
<DefaultValue("MyAccount.xsl")>
Public override Property DisplayXslt() As string
```

MyAccount.EkItem Property

Provides access to the CustomerData (see page 1478) object.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Browsable(false)]
[Description("Returns the current CustomerData object")]
[Personalizable()]
public Ektron.Cms.Commerce.CustomerData EkItem;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Browsable(false)>
<Description("Returns the current CustomerData object")>
<Personalizable()>
Public Property EkItem() As Ektron.Cms.Commerce.CustomerData
```

MyAccount.LoadingImage Property

Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")]
```

```
[DefaultValue("ajax-loader_circle_lg.gif")]
public string LoadingImage;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")>
<DefaultValue("ajax-loader_circle_lg.gif")>
Public Property LoadingImage() As string
```

MyAccount.RedirectUrl Property

Enter the path of a template that contains the Login (see page 347) server control. When a site visitor arrives at this control and is not logged in, the control checks this property for a path to template containing a Login (see page 347) server control. If one exists, the site visitor is sent to that template. Once he's logged in, he's returned to the template containing the MyAccount (see page 409) server control. If no path exists, the control checks the following key the web.config file for a path.

```
<add key="ek_RedirectToLoginURL" value="" />
```

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("This value may be set to point to a login page, when an unauthenticated visitor
is detected.")]
[DefaultValue("")]
public string RedirectUrl;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("This value may be set to point to a login page, when an unauthenticated visitor
is detected.")>
<DefaultValue("")>
Public Property RedirectUrl() As string
```

MyAccount.Stylesheet Property

Specify the path to a style sheet for use with this server control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with this server control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[TypeConverter(typeof(MyAccountCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/MyAccount.css")]
public string Stylesheet;
```

Visual Basic

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with this server control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<TypeConverter(typeof(MyAccountCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/MyAccount.css")>
Public Property Stylesheet() As string
```

MyAccount.UseMsAjax Property

When true, uses Microsoft Ajax.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
```

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("If true, uses Microsoft Ajax to perform callback. Otherwise uses Ektron Library
callback (default).")]
[DefaultValue("false")]
public bool UseMsAjax;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("If true, uses Microsoft Ajax to perform callback. Otherwise uses Ektron Library
callback (default).")>
<DefaultValue("false")>
Public Property UseMsAjax() As bool
```

OrderList Class

Summary description for OrderList control

Class Hierarchy

C#

```
[DefaultProperty("DefaultUserID"), ToolboxData("<{0}:OrderList runat=server></{0}:OrderList>")]
public class OrderList : ECommerceControl, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultUserID"), ToolboxData("<{0}:OrderList runat=server></{0}:OrderList>")>
Public Class OrderList
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

File

OrderList.cs

OrderList Methods

The methods of the OrderList class are listed here.

Public Methods

	Name	Description
	Fill (see page 414)	Populates the data for the Favorites (see page 236).
	ToString (see page 415)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 415)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

OrderList.Fill Method

Populates the data for the Favorites (see page 236).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

OrderList.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

OrderList.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

OrderList Properties

The properties of the OrderList class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 415)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, set the CacheInterval property to 300 (three hundred).
	DefaultUserID (see page 416)	The default ID of the user for which to display a list of orders.
	DisplayXslt (see page 416)	Specify an external XSLT file to render the control. By default, the control uses OrderList.xsl. This file is located in:
	DynamicOrderParameter (see page 417)	Sets the QueryString parameter name which is used to pass an order ID dynamically. By default, this parameter is OID.
	DynamicProductParameter (see page 417)	The QueryString parameter name which is used to pass the product ID to the product details page. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors click a product in their order list, this parameter is passed with the product's ID to the product details page.
	EkItem (see page 417)	Provides access to List of OrderData (see page 1520).
	PageSize (see page 418)	Specifies the number of orders to show per page.
	Stylesheet (see page 418)	Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.
	TemplateProduct (see page 418)	The URL path of the template that allows a site visitor to view the details of a product in an order. When a path is entered, the product's title becomes a link that allows site visitors to navigate to the template containing details of the product; for example, the template containing a Product (see page 431) server control.

OrderList.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public new virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public new virtual Property CacheInterval() As double
```

OrderList.DefaultUserID Property

The default ID of the user for which to display a list of orders.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The default ID of the user for which to display a list of orders.")]
public long DefaultUserID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The default ID of the user for which to display a list of orders.")>
Public Property DefaultUserID() As long
```

OrderList.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses OrderList.xsl. This file is located in:

```
[SiteRoot]\Workarea\Xslt\Commerce
```

Important: If you want to edit this file, it is strongly recommended that you make a copy, change its name and move it outside of the Workarea folder. If you make changes to this file and do not move it out of the Workarea, this file and the changes will be lost when you upgrade.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses OrderList.xsl. This file is located in: [SiteRoot]\\Workarea\\Xslt\\Commerce")]
[TypeConverter(typeof(XsltOrderTypeConverter))]
[DefaultValue("OrderList.xsl")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
```

```
<Description("Specify an external XSLT file to render the control. By default, the control
uses OrderList.xsl. This file is located in: [SiteRoot]\\Workarea\\Xslt\\Commerce")>
<TypeConverter(typeof(XsltOrderTypeConverter))>
<DefaultValue("OrderList.xsl")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

OrderList.DynamicOrderParameter Property

Sets the QueryString parameter name which is used to pass an order ID dynamically. By default, this parameter is OID.

C#

```
[Category("Ektron")]
[DefaultValue("oid")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Sets the QueryString parameter name which is used to pass an order ID
dynamically. By default, this parameter is OID.")]
public string DynamicOrderParameter;
```

Visual Basic

```
<Category("Ektron")>
<DefaultValue("oid")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Sets the QueryString parameter name which is used to pass an order ID
dynamically. By default, this parameter is OID.")>
Public Property DynamicOrderParameter() As string
```

OrderList.DynamicProductParameter Property

The QueryString parameter name which is used to pass the product ID to the product details page. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors click a product in their order list, this parameter is passed with the product's ID to the product details page.

C#

```
[Category("Ektron")]
[DefaultValue("id")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("The QueryString parameter name which is used to pass the product ID to the
product details page. For example, if your QueryString parameter for products is PID, enter
that in this property. Then, when site visitors click a product in their order list, this
parameter is passed with the product?s ID to the product details page.")]
public string DynamicProductParameter;
```

Visual Basic

```
<Category("Ektron")>
<DefaultValue("id")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("The QueryString parameter name which is used to pass the product ID to the
product details page. For example, if your QueryString parameter for products is PID, enter
that in this property. Then, when site visitors click a product in their order list, this
parameter is passed with the product?s ID to the product details page.")>
Public Property DynamicProductParameter() As string
```

OrderList.EkItem Property

Provides access to List of OrderData (see page 1520).

C#

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Provides access to List of OrderData.")]
[Personalizable()]
```

```
public List<OrderData> EkItem;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Provides access to List of OrderData.")>
<Personalizable()>
Public Property EkItem() As List<OrderData>
```

OrderList.PageSize Property

Specifies the number of orders to show per page.

C#

```
[Category("Ektron")]
[DefaultValue(10)]
[TypeConverter(typeof(int))]
[Description("Specifies the number of orders to show per page")]
public int PageSize;
```

Visual Basic

```
<Category("Ektron")>
<DefaultValue(10)>
<TypeConverter(typeof(int))>
<Description("Specifies the number of orders to show per page")>
Public Property PageSize() As Integer
```

OrderList.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[TypeConverter(typeof(OrderListCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/OrderList.css")]
public string Stylesheet;
```

Visual Basic

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<TypeConverter(typeof(OrderListCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/OrderList.css")>
Public Property Stylesheet() As string
```

OrderList.TemplateProduct Property

The URL path of the template that allows a site visitor to view the details of a product in an order. When a path is entered, the product's title becomes a link that allows site visitors to navigate to the template containing details of the product; for example, the template containing a Product (see page 431) server control.

C#

```
[Category("Ektron")]
[DefaultValue("")]
[Description("The URL path of the template that allows a site visitor to view the details of a
product in an order. When a path is entered, the product?s title becomes a link that allows
site visitors to navigate to the template containing details of the product; for example, the
template containing a Product server control.")]
public string TemplateProduct;
```

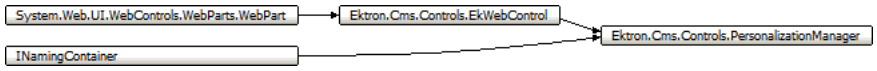
Visual Basic

```
<Category("Ektron")>
<DefaultValue("")>
<Description("The URL path of the template that allows a site visitor to view the details of a
product in an order. When a path is entered, the product?s title becomes a link that allows
site visitors to navigate to the template containing details of the product; for example, the
template containing a Product server control.")>
Public Property TemplateProduct() As string
```

PersonalizationManager Class

The PersonalizationManager server control allows site users to personalize, add, edit and hide content on a Web page. The level of privileges assigned to a site user in CMS400.NET determines what personalization functionality is available when the control is active. The Personalization (see page 11) manager only appears when a user is logged in to the site.

Class Hierarchy



C#

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:PersonalizationManager runat=server></{0}:PersonalizationManager>")]
public class PersonalizationManager : EkWebControl, INamingContainer;
```

Visual Basic

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:PersonalizationManager runat=server></{0}:PersonalizationManager>")>
Public Class PersonalizationManager
Inherits EkWebControl
Inherits INamingContainer
```

File

PersonalizationManager.cs

See Also

The Administrator Manual section "Personalization (see page 11) Feature" and the Developer Manual section "Personalization (see page 11) Server Controls (see page 7)" > "PersonalizationManager Server Control".

PersonalizationManager Methods

The methods of the PersonalizationManager class are listed here.

Public Methods

	Name	Description
	Fill (see page 419)	Method that can be overridden and used to grab data from the back end for each server control.

PersonalizationManager.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

PersonalizationManager Properties

The properties of the PersonalizationManager class are listed here.

Public Properties

	Name	Description
	AddContentExitText (see page 420)	Gets or sets the text for the Add Content (see page 822) Exit link. The default text is **Done Add Content (see page 822)**.
	AddContentText (see page 420)	Gets or sets the text for the Add Content (see page 822) link. The default is **Add Content (see page 822).**
	CurrentUserButtonText (see page 421)	Gets or sets the text for the Current User (see page 950) radio button. The default is **Current User (see page 950).**
	Display (see page 421)	Gets or sets the display format to vertical or horizontal for the Personalization (see page 11) Manager. The default is Vertical.
	HeaderBackColor (see page 421)	`HeaderBackColor` sets the background color of the personalization manager header.
	HeaderText (see page 422)	The text that appears in the header of the Personalization (see page 11) Manager.
	PersonalizeExitText (see page 422)	Gets or sets the text for the Exit link. The default is **Done**.
	PersonalizeScopeText (see page 422)	Gets or sets the text for the Scope. The default is **Personalization (see page 11) For.**
	PersonalizeStartText (see page 423)	Gets or sets the text for the Start link. The default is **Personalize.**
	PublicUserButtonText (see page 423)	Gets or sets the text for the Public User (see page 950) radio button. The default is **Public User (see page 950).**
	ResetLinkText (see page 423)	Gets or sets the text for the Reset link text. The default is **Reset To Default.**
	ResetLinkToolTip (see page 424)	Gets or sets the tool tip text for the Reset link. The default is **Reset the current user's personalization data for the page.**
	UrlCompareMode (see page 424)	Please select Path | PathAndQuery

PersonalizationManager.AddContentExitText Property

Gets or sets the text for the Add Content (see page 822) Exit link. The default text is **Done Add Content (see page 822)**.

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_ADDEXIT)]
[Description("Set text for exit add content link. Default value is set as \"" +
EkWebControl.RES_PMC_ADDEXIT + "\".")]
public string AddContentExitText;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_ADDEXIT)>
<Description("Set text for exit add content link. Default value is set as \"" +
EkWebControl.RES_PMC_ADDEXIT + "\".")>
Public Property AddContentExitText() As string
```

PersonalizationManager.AddContentText Property

Gets or sets the text for the Add Content (see page 822) link. The default is **Add Content (see page 822).**

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
```

```
[DefaultValue(EkWebControl.RES_PMC_ADD)]
[Description("Set text for add content link. Default value is set as \"" +
EkWebControl.RES_PMC_ADD + "\".")]
public string AddContentText;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_ADD)>
<Description("Set text for add content link. Default value is set as \"" +
EkWebControl.RES_PMC_ADD + "\".")>
Public Property AddContentText() As string
```

PersonalizationManager.CurrentUserButtonText Property

Gets or sets the text for the Current User (see page 950) radio button. The default is **Current User (see page 950).**

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_USRSCOPE)]
[Description("The display text for the current user radio button. Default value is set as \""
+ EkWebControl.RES_PMC_USRSCOPE + "\".")]
public string CurrentUserButtonText;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_USRSCOPE)>
<Description("The display text for the current user radio button. Default value is set as \""
+ EkWebControl.RES_PMC_USRSCOPE + "\".")>
Public Property CurrentUserButtonText() As string
```

PersonalizationManager.Display Property

Gets or sets the display format to vertical or horizontal for the Personalization (see page 11) Manager. The default is Vertical.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(DisplayFormat.Vertical)]
[Description("The display format of the image - horizontal or vertical.")]
public DisplayFormat Display;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(DisplayFormat.Vertical)>
<Description("The display format of the image - horizontal or vertical.")>
Public Property Display() As DisplayFormat
```

PersonalizationManager.HeaderBackColor Property

`HeaderBackColor` sets the background color of the personalization manager header.

C#

```
[Description("HeaderBackColor sets the header background color."), Category("Ektron")]
[Bindable(false)]
[Browsable(true)]
[TypeConverter(typeof(WebColorConverter))]
public Color HeaderBackColor;
```

Visual Basic

```
<Description("HeaderBackColor sets the header background color."), Category("Ektron")>
<Bindable(false)>
<Browsable(true)>
<TypeConverter(typeof(WebColorConverter))>
Public Property HeaderBackColor() As Color
```

PersonalizationManager.HeaderText Property

The text that appears in the header of the Personalization (see page 11) Manager.

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_HEADER)]
[Description("The HeaderText of the control. Default value is set as \"" +
EkWebControl.RES_PMC_HEADER + "\".")]
public string HeaderText;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_HEADER)>
<Description("The HeaderText of the control. Default value is set as \"" +
EkWebControl.RES_PMC_HEADER + "\".")>
Public Property HeaderText() As string
```

PersonalizationManager.PersonalizeExitText Property

Gets or sets the text for the Exit link. The default is **Done**.

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_EXIT)]
[Description("Set text for exit personalization mode link. Default value is set as \"" +
EkWebControl.RES_PMC_EXIT + "\".")]
public string PersonalizeExitText;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_EXIT)>
<Description("Set text for exit personalization mode link. Default value is set as \"" +
EkWebControl.RES_PMC_EXIT + "\".")>
Public Property PersonalizeExitText() As string
```

PersonalizationManager.PersonalizeScopeText Property

Gets or sets the text for the Scope. The default is **Personalization (see page 11) For.**

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_SCPTXT)]
[Description("Set legend text for Personalize Scope fields. Default value is set as \"" +
EkWebControl.RES_PMC_SCPTXT + "\".")]
public string PersonalizeScopeText;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_SCPTXT)>
<Description("Set legend text for Personalize Scope fields. Default value is set as \"" +
EkWebControl.RES_PMC_SCPTXT + "\".")>
Public Property PersonalizeScopeText() As string
```

PersonalizationManager.PersonalizeStartText Property

Gets or sets the text for the Start link. The default is **Personalize.**

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_START)]
[Description("Set text for start personalize link. Default value is set as \"" +
EkWebControl.RES_PMC_START + "\".")]
public string PersonalizeStartText;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_START)>
<Description("Set text for start personalize link. Default value is set as \"" +
EkWebControl.RES_PMC_START + "\".")>
Public Property PersonalizeStartText() As string
```

PersonalizationManager.PublicUserButtonText Property

Gets or sets the text for the Public User (see page 950) radio button. The default is **Public User (see page 950).**

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_SHRSCOPE)]
[Description("The display text for the public user radio button. Default value is set as \""
+EkWebControl.RES_PMC_SHRSCOPE + "\".")]
public string PublicUserButtonText;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_SHRSCOPE)>
<Description("The display text for the public user radio button. Default value is set as \""
+EkWebControl.RES_PMC_SHRSCOPE + "\".")>
Public Property PublicUserButtonText() As string
```

PersonalizationManager.ResetLinkText Property

Gets or sets the text for the Reset link text. The default is **Reset To Default.**

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_RESET)]
[Description("The display text for the reset link. Default value is set as \"" +
```

```
EkWebControl.RES_PMC_RESET + "\".")]
public string ResetLinkText;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_RESET)>
<Description("The display text for the reset link. Default value is set as \"" +
EkWebControl.RES_PMC_RESET + "\".")>
Public Property ResetLinkText() As string
```

PersonalizationManager.ResetLinkToolTip Property

Gets or sets the tool tip text for the Reset link. The default is **Reset the current user's personalization data for the page.**

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_RESET_ALT)]
[Description("The display text for the reset link. Default value is set as \"" +
EkWebControl.RES_PMC_RESET_ALT + ".\".")]
public string ResetLinkToolTip;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_RESET_ALT)>
<Description("The display text for the reset link. Default value is set as \"" +
EkWebControl.RES_PMC_RESET_ALT + ".\".")>
Public Property ResetLinkToolTip() As string
```

PersonalizationManager.UrlCompareMode Property

Please select Path | PathAndQuery

C#

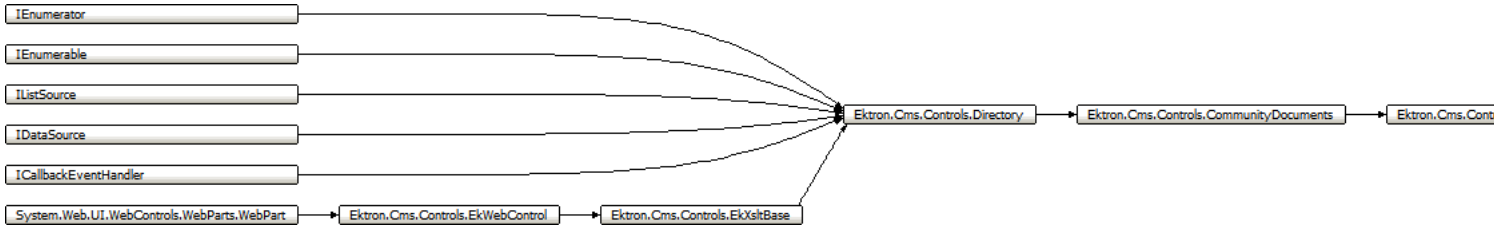
```
[Bindable(false)]
[Category("Ektron")]
[Description("")]
public string UrlCompareMode;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("")>
Public Property UrlCompareMode() As string
```

PhotoGallery Class

Class Hierarchy



C#

```
[DefaultProperty("DefaultUserID")]
[ToolboxData("<{0}:PhotoGallery runat=server></{0}:PhotoGallery>")]
public class PhotoGallery : CommunityDocuments;
```

Visual Basic

```
<DefaultProperty("DefaultUserID")>
<ToolboxData("<{0}:PhotoGallery runat=server></{0}:PhotoGallery>")>
Public Class PhotoGallery
Inherits CommunityDocuments
```

File

PhotoGallery.cs

Description

This is class Ektron.Cms.Controls.PhotoGallery.

PhotoGallery Methods

The methods of the PhotoGallery class are listed here.

Public Methods

	Name	Description
	Fill (see page 425)	Populates the data for the Taxonomy (see page 876).

PhotoGallery.Fill Method

Populates the data for the Taxonomy (see page 876).

C#

```
public override void Fill();
```

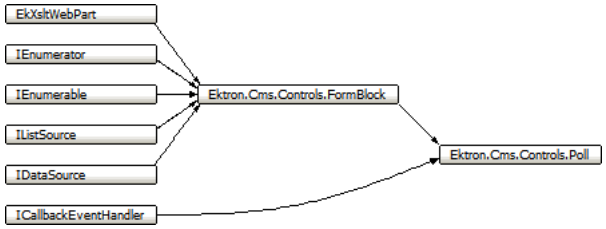
Visual Basic

```
Public override Function Fill() As void
```

Poll Class

The Poll Server Control displays a poll or survey created from an Ektron CMS400.NET form on a Web page.

Class Hierarchy



C#

```
[DefaultProperty("DefaultFormID")]
[ToolboxData("<{0}:Poll runat=server></{0}:Poll>")]
[ToolboxBitmap("Poll.bmp")]
public class Poll : Ektron.Cms.Controls.FormBlock, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultFormID")>
<ToolboxData("<{0}:Poll runat=server></{0}:Poll>")>
<ToolboxBitmap("Poll.bmp")>
Public Class Poll
Inherits Ektron.Cms.Controls.FormBlock
Inherits ICallbackEventHandler
```

File

Poll.cs

Poll Methods

The methods of the Poll class are listed here.

Public Methods

	Name	Description
	Fill (see page 426)	Fills in the poll area with the content information from the content database. The form data is retrieved and displayed. Only use if the content must be refreshed after an form ID change or other operation that will require regenerating the display.
	WrapPoll (see page 426)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Poll.Fill Method

Fills in the poll area with the content information from the content database. The form data is retrieved and displayed. Only use if the content must be refreshed after an form ID change or other operation that will require regenerating the display.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Poll.WrapPoll Method

Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public string WrapPoll(string Html, FormData formData);
```

Visual Basic

```
Public Function WrapPoll(Html As string, formData As FormData) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

Poll Properties

The properties of the Poll class are listed here.

Public Properties

	Name	Description
	EnableAjax (see page 427)	Determines if the results will show in the area of the page that contains the poll. If the value is 'true' then the results will be shown in the area containing the poll without any modification or refresh of the surrounding contents. Otherwise, the poll results are displayed in place of the page content.
	PollID (see page 427)	Gets or Sets the Default FormID for this Control.

Poll.EnableAjax Property

Determines if the results will show in the area of the page that contains the poll. If the value is 'true' then the results will be shown in the area containing the poll without any modification or refresh of the surrounding contents. Otherwise, the poll results are displayed in place of the page content.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Browsable(false)]
[Obsolete("No longer supported. It is always defaults to true.")]
[Description("Whether the results shown be displayed in the poll location without disturbing
the surrounding contents.")]
public bool EnableAjax;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Browsable(false)>
<Obsolete("No longer supported. It is always defaults to true.")>
<Description("Whether the results shown be displayed in the poll location without disturbing
the surrounding contents.")>
Public Property EnableAjax() As bool
```

Poll.PollID Property

Gets or Sets the Default FormID for this Control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[ParenthesizePropertyName(true)]
[Editor(typeof(PollUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("The ID of the Poll to display.")]
[Personalizable(), WebBrowsable()]
public long PollID;
```

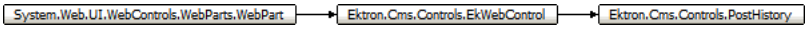
Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<ParenthesizePropertyName(true)>
<Editor(typeof(PollUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("The ID of the Poll to display.")>
<Personalizable(), WebBrowsable()>
Public Property PollID() As long
```

PostHistory Class

The PostHistory server control displays a list of posts for a given user. The posts are displayed in order by date and contain the following: Topic - the topic that contains the post Posted - the date the post was created Content (see page 822) - the content with in the post

Class Hierarchy



C#

```
[DefaultProperty("BoardID"), ToolboxData("<{0}:PostHistory runat=server></{0}:PostHistory>")]
public class PostHistory : EkWebControl;
```

Visual Basic

```
<DefaultProperty("BoardID"), ToolboxData("<{0}:PostHistory runat=server></{0}:PostHistory>")>
Public Class PostHistory
Inherits EkWebControl
```

File

PostHistory.cs

PostHistory Methods

The methods of the PostHistory class are listed here.

Public Methods

	Name	Description
	Fill (see page 428)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 428)	Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.
	Wrap (see page 429)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

PostHistory.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

PostHistory.ToString Method

Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

PostHistory.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

PostHistory Properties

The properties of the PostHistory class are listed here.

Public Properties

	Name	Description
	BoardID (see page 429)	The ID of the discussion board from which to get a user's posts.
V	CacheInterval (see page 429)	Sets the amount of time, in seconds, the server control's data is cached. The default is 0 (zero). For example, if you want to cache the data for five minutes, you set this property to 300.
	MaxNumber (see page 430)	The maximum number of posts listed. The default is **10**.
	URLPath (see page 430)	The URL path to the page the where the Forum (see page 264) server control is located.
	UserID (see page 430)	The ID of the user for whom to get the post history.

PostHistory.BoardID Property

The ID of the discussion board from which to get a user's posts.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the discussion board from which to get a user's posts.")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BoardID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the discussion board from which to get a user's posts.")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BoardID() As long
```

PostHistory.CacheInterval Property

Sets the amount of time, in seconds, the server control's data is cached. The default is 0 (zero). For example, if you want to

cache the data for five minutes, you set this property to 300.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time, in seconds, the server control?s data is cached. The
default is 0 (zero). For example, if you want to cache the data for five minutes, you set this
property to 300.")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time, in seconds, the server control?s data is cached. The
default is 0 (zero). For example, if you want to cache the data for five minutes, you set this
property to 300.")>
<DefaultValue(0)>
Public virtual Property CacheInterval() As double
```

PostHistory.MaxNumber Property

The maximum number of posts listed. The default is **10**.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of posts to display.")]
public int MaxNumber;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("The maximum number of posts to display.")>
Public Property MaxNumber() As Integer
```

PostHistory.URLPath Property

The URL path to the page the where the Forum (see page 264) server control is located.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("The URL path to the page the where the Forum server control is located.")]
public string URLPath;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("The URL path to the page the where the Forum server control is located.")>
Public Property URLPath() As string
```

PostHistory.UserID Property

The ID of the user for whom to get the post history.

C#

```
[Bindable(true)]
[Category("Ektron")]
```

```
[DefaultValue(0)]
[Description("The ID of the user for whom to get the post history.")]
public long UserID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the user for whom to get the post history.")>
Public Property UserID() As long
```

PostLoginEventArgs Class

Holds the Post-Login (see page 347) event arguments.

Class Hierarchy

C#

```
public class PostLoginEventArgs : EventArgs;
```

Visual Basic

```
Public Class PostLoginEventArgs
Inherits EventArgs
```

File

ECommerceControl.cs

Product Class

Summary description for Product

Class Hierarchy

C#

```
[DefaultProperty("DefaultProductID"), ToolboxData("<{0}:Product runat=server></{0}:Product>")]
public class Product : ECommerceControl, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultProductID"), ToolboxData("<{0}:Product runat=server></{0}:Product>")>
Public Class Product
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

File

Product.cs

Product Methods

The methods of the Product class are listed here.

Public Methods

	Name	Description
	Fill (see page 432)	Populates the data for the Favorites (see page 236).
	RedirectToBasketPage (see page 432)	Redirects to the basket page.

	ToString (see page 432)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 432)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Product.Fill Method

Populates the data for the Favorites (see page 236).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Product.RedirectToBasketPage Method

Redirects to the basket page.

C#

```
public bool RedirectToBasketPage();
```

Visual Basic

```
Public Function RedirectToBasketPage() As bool
```

Returns

bool

Product.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Product.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

Product Properties

The properties of the Product class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 433)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DefaultProductID (see page 434)	Enter a default product's ID. This is the default item that's displayed when the template containing this control is viewed by a site visitor and a product ID is not dynamically passed. To make this server control dynamic, enter zero (0) for this property and set the DynamicProductParameter to the QueryString parameter used to pass the product ID.
	DisplayXslt (see page 434)	Specify an external XSLT file to render the control. By default, the control uses product.xsl. This file is located in:
	DynamicParameter (see page 434)	The QueryString parameter name which is used to read the product ID. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors views an item's details, the product's ID is passed to this control.
	EkItems (see page 435)	Provides access to EntryData (see page 1487) object.
	ImageGallery (see page 435)	Select the type of image gallery that's displayed with a product's information. Images that appear in the gallery are set in the Workarea's Media tab when creating or editing a product. The Gallery Display property for each image on the Media tab must be set to Yes. Choices are: • List = display gallery images • None = hide gallery images In the Image Gallery, images are displayed at their smallest size. Once a site visitor clicks an image, a larger version of it is displayed in a thick box.
	OverrideXslt (see page 436)	Specifies an XSLT identified in the Edit Product (see page 431) Type Configuration screen. Use this to specify an override for the products smart-form XSLT; 0 (default) uses the XSLT defined in the product type (e.g. PackageDisplayXslt), 1 for Xslt1, 2 for Xslt2, etc.
	ShowAddToCart (see page 436)	Set to true if you want to the Add to Cart (see page 81) button to appear. The default is True. Setting this property to false allows you to show details of an item, but not offer it for sale. For example, you have an item that is no longer for sale, but you want to allow people who purchased the item to view its details. Also, by using code behind to dynamically set the property, you could create code that looks at your inventory system and hides the button depending on whether a product is in stock.
	Stylesheet (see page 436)	Specify the path to a style sheet for use with this sever control. The location can be relative or absolute.
	TemplateCart (see page 437)	Enter the URL path of the template that contains the Cart (see page 81) server control. This path can be relative or absolute.

Product.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public new virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public new virtual Property CacheInterval() As double
```

Product.DefaultProductID Property

Enter a default product's ID. This is the default item that's displayed when the template containing this control is viewed by a site visitor and a product ID is not dynamically passed. To make this server control dynamic, enter zero (0) for this property and set the DynamicProductParameter to the QueryString parameter used to pass the product ID.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enter a default product?s ID. This is the default item that?s displayed when the
template containing this control is viewed by a site visitor and a product ID is not
dynamically passed. To make this server control dynamic, enter zero (0) for this property and
set the DynamicProductParameter to the QueryString parameter used to pass the product ID. ")]
public long DefaultProductID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Enter a default product?s ID. This is the default item that?s displayed when the
template containing this control is viewed by a site visitor and a product ID is not
dynamically passed. To make this server control dynamic, enter zero (0) for this property and
set the DynamicProductParameter to the QueryString parameter used to pass the product ID. ")>
Public Property DefaultProductID() As long
```

Product.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses product.xsl. This file is located in:

```
[SiteRoot]\Workarea\Xslt\Commerce
```

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses product.xsl. This file is located in [SiteRoot]\\Workarea\\Xslt\\Commerce")]
[TypeConverter(typeof(XsltProductTypeConverter))]
[DefaultValue("Product.xsl")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses product.xsl. This file is located in [SiteRoot]\\Workarea\\Xslt\\Commerce")>
<TypeConverter(typeof(XsltProductTypeConverter))>
<DefaultValue("Product.xsl")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

Product.DynamicParameter Property

The QueryString parameter name which is used to read the product ID. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors views an item's details, the product's ID is passed to this control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("id")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
```

```
[Description("The QueryString parameter name which is used to read the product ID. For
example, if your QueryString parameter for products is PID, enter that in this property. Then,
when site visitors views an item?s details, the product?s ID is passed to this control.")]
[Personalizable()]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("id")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("The QueryString parameter name which is used to read the product ID. For
example, if your QueryString parameter for products is PID, enter that in this property. Then,
when site visitors views an item?s details, the product?s ID is passed to this control.")>
<Personalizable()>
Public Property DynamicParameter() As string
```

Product.EkItems Property

Provides access to EntryData (see page 1487) object.

C#

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Provides access to EntryData object.")]
public EntryData EkItems;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Provides access to EntryData object.")>
Public Property EkItems() As EntryData
```

Product.ImageGallery Property

Select the type of image gallery that's displayed with a product's information. Images that appear in the gallery are set in the Workarea's Media tab when creating or editing a product. The Gallery Display property for each image on the Media tab must be set to Yes. Choices are:

- List = display gallery images
- None = hide gallery images

In the Image Gallery, images are displayed at their smallest size. Once a site visitor clicks an image, a larger version of it is displayed in a thick box.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(ImageGalleryType.List)]
[Description("elect the type of image gallery that?s displayed with a product?s information.
Images that appear in the gallery are set in the Workarea?s Media tab when creating or editing
a product.")]
public ImageGalleryType ImageGallery;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(ImageGalleryType.List)>
<Description("elect the type of image gallery that?s displayed with a product?s information.
Images that appear in the gallery are set in the Workarea?s Media tab when creating or editing
a product.")>
```

```
Public Property ImageGallery() As ImageGalleryType
```

Product.OverrideXslt Property

Specifies an XSLT identified in the Edit Product (see page 431) Type Configuration screen. Use this to specify an override for the products smart-form XSLT; 0 (default) uses the XSLT defined in the product type (e.g. PackageDisplayXslt), 1 for Xslt1, 2 for Xslt2, etc.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Specifies an XSLT identified in the Edit Product Type Configuration screen. Used
to override the products smart-form XSLT; 0 (default) uses the XSLT defined in the product
type (e.g. PackageDisplayXslt), 1 for Xslt1, 2 for Xslt2, etc. Note that this XSLT is run on
only the smart-forms markup, and the results are then available for the DisplayXslt to
transform, as children of node <markupTransformed>.")]
public int OverrideXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Specifies an XSLT identified in the Edit Product Type Configuration screen. Used
to override the products smart-form XSLT; 0 (default) uses the XSLT defined in the product
type (e.g. PackageDisplayXslt), 1 for Xslt1, 2 for Xslt2, etc. Note that this XSLT is run on
only the smart-forms markup, and the results are then available for the DisplayXslt to
transform, as children of node <markupTransformed>.")>
Public Property OverrideXslt() As Integer
```

Product.ShowAddToCart Property

Set to true if you want to the Add to Cart (see page 81) button to appear. The default is True. Setting this property to false allows you to show details of an item, but not offer it for sale. For example, you have an item that is no longer for sale, but you want to allow people who purchased the item to view its details. Also, by using code behind to dynamically set the property, you could create code that looks at your inventory system and hides the button depending on whether a product is in stock.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true if you want to the Add to Cart button to appear. The default is
True. Setting this property to false allows you to show details of an item, but not offer it
for sale.")]
public bool ShowAddToCart;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true if you want to the Add to Cart button to appear. The default is
True. Setting this property to false allows you to show details of an item, but not offer it
for sale.")>
Public Property ShowAddToCart() As bool
```

Product.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute.

C#

```
[Category("Ektron")]
[Description("Allows selecting a different CSS file for the Product server control.")]
```

```
[TypeConverter(typeof(ProductCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/Product.css")]
public string Stylesheet;
```

Visual Basic

```
<Category("Ektron")>
<Description("Allows selecting a different CSS file for the Product server control.")>
<TypeConverter(typeof(ProductCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/Product.css")>
Public Property Stylesheet() As string
```

Product.TemplateCart Property

Enter the URL path of the template that contains the Cart (see page 81) server control. This path can be relative or absolute.

C#

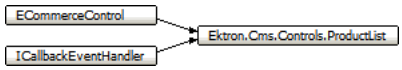
```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Enter the URL path of the template that contains the Cart server control. This
path can be relative or absolute. ")]
public string TemplateCart;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Enter the URL path of the template that contains the Cart server control. This
path can be relative or absolute. ")>
Public Property TemplateCart() As string
```

ProductList Class

Class Hierarchy



C#

```
[DefaultProperty("DefaultProductListID"), ToolboxData("<{0}:ProductList
runat=server></{0}:ProductList>")]
public class ProductList : ECommerceControl, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultProductListID"), ToolboxData("<{0}:ProductList
runat=server></{0}:ProductList>")>
Public Class ProductList
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

File

ProductList.cs

Description

This is class Ektron.Cms.Controls.ProductList.

ProductList Methods

The methods of the ProductList class are listed here.

Public Methods

	Name	Description
	Fill (see page 438)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 438)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 438)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

ProductList.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

ProductList.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

ProductList.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

ProductList Properties

The properties of the ProductList class are listed here.

Public Properties

	Name	Description
	DisplayXslt (see page 439)	Specify an external XSLT file to render the control. By default, the control uses ProductList.xsl. This file is located in:
	DynamicParameter (see page 440)	Sets the QueryString parameter to read a source's ID dynamically. For example, if your QueryString parameter for a source ID is SID, enter that.
	EkItems (see page 440)	Returns a List of Product (see page 431) Entry Data.
	IdList (see page 440)	Enter a comma separated list of IDs when the SourceType (see page 443) property is set to IdList, TaxonomyList or CatalogList.

	LoadingImage (see page 441)	Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif**.
	PageSize (see page 441)	Specify the number of items to show per page when a site visitor views the product list. If the number of items exceeds the quantity defined in this property, the list is paged and the site visitor can use the paging navigation system to move through the list.
	PageSpan (see page 441)	The number of pages to show before and after the current page. Enter zero to show all pages.
	ProductCriteria (see page 442)	Allows a developer to set the criteria for getting products from CMS400.NET.
	Recursive (see page 442)	When SourceType (see page 443) is set to SourceObjectType.Catalog, setting Recursive to true will cause the control to include products from all child catalogs contained within the catalog specified by the supplied catalog ID.
	SortMode (see page 442)	Select the default way the product list is sorted when a site visitor first views the page. Once the page is loaded, a site visitor can sort the list from a drop down list. Choices are: • **Title** - sorts in alphabetical order. • **SKU** - sorts by the product number. This number is typically a unique number supplied by the producer of the item. • **Price Descending** - sorts by price from highest to lowest. • **Price Ascending** - sorts by price from lowest to highest.
	SourceId (see page 443)	The ID of the catalog or taxonomy that is being used as a product list. This property is used when the SourceType (see page 443) property is set to Catalog or Taxonomy (see page 876).
	SourceType (see page 443)	The type of source being used to create the list. Choices are: • **Catalog** - use the SourceId (see page 443) property to specify the ID of a single catalog when this source type is selected. • **CatalogList** - use the IdList (see page 440) property to specify a list of catalog IDs when this source type is selected. • **Taxonomy (see page 876)** - use the SourceId (see page 443) property to specify the ID of the single taxonomy when this source type is selected. • **TaxonomyList** - Use IdList (see page 440) property to specify a list of taxonomy IDs when this type is selected. • **IdList (see page 440)** - Use the IdList (see page 440) property to specify a list of product... more (see page 443)
	Stylesheet (see page 444)	Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.
	TaxonomyDepth (see page 444)	Enter the number of taxonomy levels to retrieve below each taxonomy/category. Only the top level is displayed in the control. The rest of the levels are available through XML in code behind. For example, if the taxonomy is DVDs > Movies > Comedies , and you set Taxonomy (see page 876) Depth to 2, only DVDs and Movie are available in code behind. To retrieve the all of the categories for a taxonomy recursively, enter -1. A depth greater than 1 or using -1 is only useful if you create a custom output using the Taxonomy (see page 876)'s XML. The default value is 1.... more (see page 444)
	TemplateProduct (see page 445)	Specify the URL that contains the Product (see page 431) server control. This allows a site visitor to see the details of the product when the product link is clicked.
	UseMsAjax (see page 445)	When true, uses Microsoft Ajax.

ProductList.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses ProductList.xsl. This file is located in:

```
[SiteRoot]\Workarea\Xslt\Commerce
```

Important: If you want to edit this file, it is strongly recommended that you make a copy, change its name and move it outside of the Workarea folder. If you make changes to this file and do not move it out of the Workarea, this file and the changes will be lost when you upgrade.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses ProductList.xsl. This file is located in [SiteRoot]\\Workarea\\Xslt\\Commerce")]
```

```
[TypeConverter(typeof(XsltProductListTypeConverter))]
[DefaultValue("ProductList.xsl")]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses ProductList.xsl. This file is located in [SiteRoot]\\Workarea\\Xslt\\Commerce")>
<TypeConverter(typeof(XsltProductListTypeConverter))>
<DefaultValue("ProductList.xsl")>
Public override Property DisplayXslt() As string
```

ProductList.DynamicParameter Property

Sets the QueryString parameter to read a source's ID dynamically. For example, if your QueryString parameter for a source ID is SID, enter that.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("Sets the QueryString parameter to read a source?s ID dynamically. For example,
if your QueryString parameter for a source ID is SID, enter that.")]
[DefaultValue("")]
public string DynamicParameter;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("Sets the QueryString parameter to read a source?s ID dynamically. For example,
if your QueryString parameter for a source ID is SID, enter that.")>
<DefaultValue("")>
Public Property DynamicParameter() As string
```

ProductList.EkItems Property

Returns a List of Product (see page 431) Entry Data.

C#

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Returns a List of Product Entry Data.")]
[Personalizable()]
public List<EntryData> EkItems;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Returns a List of Product Entry Data.")>
<Personalizable()>
Public Property EkItems() As List<EntryData>
```

ProductList.IdList Property

Enter a comma separated list of IDs when the SourceType (see page 443) property is set to IdList, TaxonomyList or CatalogList.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
```

```
[Description("Enter a comma separated list of IDs when the SourceType property is set to
IdList, TaxonomyList or CatalogList.")]
[Personalizable(), WebBrowsable()]
public string IdList;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("Enter a comma separated list of IDs when the SourceType property is set to
IdList, TaxonomyList or CatalogList.")>
<Personalizable(), WebBrowsable()>
Public Property IdList() As string
```

ProductList.LoadingImage Property

Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif**.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")]
[DefaultValue("ajax-loader_circle_lg.gif")]
public string LoadingImage;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")>
<DefaultValue("ajax-loader_circle_lg.gif")>
Public Property LoadingImage() As string
```

ProductList.PageSize Property

Specify the number of items to show per page when a site visitor views the product list. If the number of items exceeds the quantity defined in this property, the list is paged and the site visitor can use the paging navigation system to move through the list.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("Specify the number of items to show per page when a site visitor views the
product list. If the number of items exceeds the quantity defined in this property, the list
is paged and the site visitor can use the paging navigation system to move through the list.")]
[DefaultValue(0)]
public int PageSize;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("Specify the number of items to show per page when a site visitor views the
product list. If the number of items exceeds the quantity defined in this property, the list
is paged and the site visitor can use the paging navigation system to move through the list.")>
<DefaultValue(0)>
Public Property PageSize() As Integer
```

ProductList.PageSpan Property

The number of pages to show before and after the current page. Enter zero to show all pages.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("The number of pages to show before and after the current page. Enter zero to
show all pages.")]
[DefaultValue(0)]
public int PageSpan;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("The number of pages to show before and after the current page. Enter zero to
show all pages.")>
<DefaultValue(0)>
Public Property PageSpan() As Integer
```

ProductList.ProductCriteria Property

Allows a developer to set the criteria for getting products from CMS400.NET.

C#

```
[Browsable(false)]
[Category("Ektron")]
public Criteria<EntryProperty> ProductCriteria;
```

Visual Basic

```
<Browsable(false)>
<Category("Ektron")>
Public Property ProductCriteria() As Criteria<EntryProperty>
```

ProductList.Recursive Property

When SourceType (see page 443) is set to SourceObjectType.Catalog, setting Recursive to true will cause the control to include products from all child catalogs contained within the catalog specified by the supplied catalog ID.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("When SourceType is set to SourceObjectType.Catalog, setting Recursive to true
will cause the control to include products from all child catalogs contained within the
catalog specified by the supplied catalog ID.")]
[DefaultValue("false")]
public bool Recursive;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("When SourceType is set to SourceObjectType.Catalog, setting Recursive to true
will cause the control to include products from all child catalogs contained within the
catalog specified by the supplied catalog ID.")>
<DefaultValue("false")>
Public Property Recursive() As bool
```

ProductList.SortMode Property

Select the default way the product list is sorted when a site visitor first views the page. Once the page is loaded, a site visitor can sort the list from a drop down list. Choices are:

- **Title** - sorts in alphabetical order.
- **SKU** - sorts by the product number. This number is typically a unique number supplied by the producer of the item.
- **Price Descending** - sorts by price from highest to lowest.

- **Price Ascending** - sorts by price from lowest to highest.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(SortModes))]
[Description("Select the default way the product list is sorted when a site visitor first
views the page. Once the page is loaded, a site visitor can sort the list from a drop down
list. Choices are: Title, SKU, Price Ascending and Price Descending")]
[DefaultValue("SortModes.None")]
public SortModes SortMode;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(SortModes))>
<Description("Select the default way the product list is sorted when a site visitor first
views the page. Once the page is loaded, a site visitor can sort the list from a drop down
list. Choices are: Title, SKU, Price Ascending and Price Descending")>
<DefaultValue("SortModes.None")>
Public Property SortMode() As SortModes
```

ProductList.SourceId Property

The ID of the catalog or taxonomy that is being used as a product list. This property is used when the SourceType (see page 443) property is set to Catalog or Taxonomy (see page 876).

C#

```
[Category("Ektron")]
[TypeConverter(typeof(long))]
[Description("The ID of the catalog or taxonomy that is being used as a product list. This
property is used when the SourceType property is set to Catalog or Taxonomy.")]
[DefaultValue(0)]
public long SourceId;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(long))>
<Description("The ID of the catalog or taxonomy that is being used as a product list. This
property is used when the SourceType property is set to Catalog or Taxonomy.")>
<DefaultValue(0)>
Public Property SourceId() As long
```

ProductList.SourceType Property

The type of source being used to create the list. Choices are:

- **Catalog** - use the SourceId (see page 443) property to specify the ID of a single catalog when this source type is selected.
- **CatalogList** - use the IdList (see page 440) property to specify a list of catalog IDs when this source type is selected.
- **Taxonomy (see page 876)** - use the SourceId (see page 443) property to specify the ID of the single taxonomy when this source type is selected.
- **TaxonomyList** - Use IdList (see page 440) property to specify a list of taxonomy IDs when this type is selected.
- **IdList (see page 440)** - Use the IdList (see page 440) property to specify a list of product IDs when this source type is selected.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(SourceObjectType))]
[DefaultValue("SourceObjectType.Catalog")]
[Description("The type of source being used to create the list. Choices are: Catalog,
CatalogList, Taxonomy, TaxonomyList and IdList.")]
[Personalizable()]
public SourceObjectType SourceType;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(SourceObjectType))>
<DefaultValue("SourceObjectType.Catalog")>
<Description("The type of source being used to create the list. Choices are: Catalog,
CatalogList, Taxonomy, TaxonomyList and IdList.")>
<Personalizable()>
Public Property SourceType() As SourceObjectType
```

ProductList.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with this server control. The location
can be relative or absolute.")]
[TypeConverter(typeof(ProductListCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/ProductListIE.css")]
public string Stylesheet;
```

Visual Basic

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with this server control. The location
can be relative or absolute.")>
<TypeConverter(typeof(ProductListCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/ProductListIE.css")>
Public Property Stylesheet() As string
```

ProductList.TaxonomyDepth Property

Enter the number of taxonomy levels to retrieve below each taxonomy/category. Only the top level is displayed in the control. The rest of the levels are available through XML in code behind. For example, if the taxonomy is DVDs > Movies > Comedies , and you set Taxonomy (see page 876) Depth to 2, only DVDs and Movie are available in code behind. To retrieve the all of the categories for a taxonomy recursively, enter -1. A depth greater than 1 or using -1 is only useful if you create a custom output using the Taxonomy (see page 876)'s XML. The default value is 1.

Important! - For a live site, Ektron strongly recommends leaving this value at 1. Increasing this value can slow down your live Web server. However, for testing on a staging server, you can increase the depth.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("Enter the number of taxonomy levels to retrieve below each taxonomy/category.
Only the top level is displayed in the control. The rest of the levels are available through
XML in code behind.")]
[DefaultValue(0)]
public int TaxonomyDepth;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("Enter the number of taxonomy levels to retrieve below each taxonomy/category.
Only the top level is displayed in the control. The rest of the levels are available through
XML in code behind.")>
<DefaultValue(0)>
Public Property TaxonomyDepth() As Integer
```

ProductList.TemplateProduct Property

Specify the URL that contains the Product (see page 431) server control. This allows a site visitor to see the details of the product when the product link is clicked.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("Specify the URL that contains the Product server control. This allows a site
visitor to see the details of the product when the product link is clicked.")]
[DefaultValue("")]
public string TemplateProduct;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("Specify the URL that contains the Product server control. This allows a site
visitor to see the details of the product when the product link is clicked.")>
<DefaultValue("")>
Public Property TemplateProduct() As string
```

ProductList.UseMsAjax Property

When true, uses Microsoft Ajax.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("If true, uses Microsoft Ajax to perform callback. Otherwise uses Ektron Library
callback (default).")]
[DefaultValue("false")]
public bool UseMsAjax;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("If true, uses Microsoft Ajax to perform callback. Otherwise uses Ektron Library
callback (default).")>
<DefaultValue("false")>
Public Property UseMsAjax() As bool
```

ProductSearch Class

The Web Search (see page 474) server control lets you customize the behavior of the search. You place this control on any Web form from which a site visitor can search your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 725)" > "Web Site (see page 725) Search (see page 474)."

Class Hierarchy

C#

```
[DefaultProperty("CatalogId"), ToolboxData("<{0}:ProductSearch
runat=server></{0}:ProductSearch>")]
public class ProductSearch : ECommerceControl, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("CatalogId"), ToolboxData("<{0}:ProductSearch
runat=server></{0}:ProductSearch>")>
Public Class ProductSearch
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

File

ProductSearch.cs

ProductSearch.ProductSearch Constructor

Constructor for the WebSearch (see page 529) class.

C#

```
public ProductSearch();
```

Visual Basic

```
Public Sub New()
```

ProductSearch Methods

The methods of the ProductSearch class are listed here.

Public Methods

	Name	Description
	Fill (see page 446)	Method that can be overridden and used to grab data from the back end for each server control.
	GetCallbackResult (see page 446)	Returns search results to the client callback function
	RaiseCallbackEvent (see page 447)	Processes the callback event by performing the search.
	Wrap (see page 447)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

ProductSearch.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

ProductSearch.GetCallbackResult Method

Returns search results to the client callback function

C#

```
public string GetCallbackResult();
```

Visual Basic

```
Public Function GetCallbackResult() As string
```

Returns

String containing search results.

ProductSearch.RaiseCallbackEvent Method

Processes the callback event by performing the search.

C#

```
public void RaiseCallbackEvent(string eventArgs);
```

Visual Basic

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

Parameters

Parameters	Description
string eventArgs	Name-value pairs representing form fields passed by client callback function

ProductSearch.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

ProductSearch Properties

The properties of the ProductSearch class are listed here.

Public Properties

	Name	Description
	ButtonImgSrc (see page 449)	If you want to display an image on the Submit button, enter the server path to it.
	ButtonText (see page 450)	The text used for the Submit button if no image source is identified. If an image source is identified, this is alternative text for the button.
	CacheInterval (see page 450)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	CatalogId (see page 450)	The folder at which the search begins. The starting folder need not be the root folder. The **Recursive (see page 455)** property determines if the search examines this folder's subfolders.
	CustomOrderBy (see page 451)	Provide a property's Friendly Name defined in the Indexing Service to sort search results by that property. Results can be ascending or descending based on `OrderDirection (see page 455)`. If you enter an invalid property, no search results are returned. If you specify `CustomOrderBy` and `OrderBy (see page 454)`, the `OrderBy (see page 454)` property is ignored. The Friendly Name of a property can be found in **Computer Management** > **Services (see page 1083) and Applications** > **Indexing Service** > **Your Index** > **Properties** > **Friendly Name** column.
	DisplayXslt (see page 451)	Specify an external XSLT file to render the control. By default, the control uses ProductSearch.xsl. This file is located in:
	DynamicProductParameter (see page 451)	The QueryString parameter name which is used to pass the product ID to the product details page.
	EnableAdvancedLink (see page 452)	Set to **True** to display an additional tab (Advanced) on the ProductSearch (see page 445) control.
	EndCount (see page 452)	Get the numerical record of the first item on a page.

	IsInWorkArea (see page 452)	Is server control in workarea
	LinkTarget (see page 453)	Gets or sets the target window for the hyperlinks in the search result. The choices are: • **_blank** - loads the link in a new blank window. This window is not named. • **_self** - loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target. • **_parent** - loads the link in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** - loads the link in the full body of the window. This defaults to acting like "_self" if the... more (see page 453)
	LoadingImage (see page 453)	Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**.
	MaxCharacters (see page 453)	The maximum number of characters the Search (see page 474) text box accepts. If you enter a value less than 50, it is a good idea to set the TextBoxSize (see page 461) property to the same number.
	MaxTeaserLength (see page 454)	Limits the length of any returned content's abstract. To allow unlimited length, set to zero. This property is active only if both of these conditions are true. • you use the DisplayXslt (see page 451) property to identify an XSLT and ecmteaser as a value of that property. • the ShowCustomSummary (see page 458) property is set to false. If ShowCustomSummary (see page 458) is set to true, the entire summary appears in search results.
	OrderBy (see page 454)	The order of the search results. Choices are: • **Title** - The content title (alphabetical) • **ID** - The content ID number • **Date Created** - The date the content was created • **Date Modified** - The date the content was most recently modified • **Editor** - The user who last edited the content (alphabetical) • **Rank** - The rank assigned to the content. See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 725)" > "Display of Search (see page 474) Results" > "Search (see page 474) Result Ranking"
	OrderDirection (see page 455)	The direction in which search results are sorted. The default is Ascending. • **Ascending** - Alphabetical results from A to Z; numeric values low to high; dates from oldest to most recent • **Descending** - Alphabetical results from Z to A; numeric values high to low; dates from most recent to oldest
	Recursive (see page 455)	Set to True to search sub-folders of the starting folder. The starting folder is identified in the `CatalogID` property.
	ResultCount (see page 455)	The number of search results. This is a read-only property.
V	ResultDataTable (see page 456)	Allows a developer to get results from the search control as a DataTable
	ResultsPageSize (see page 456)	An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary is truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT.
	ResultTagId (see page 456)	Lets you designate where search results appear. You can place search criteria in one area of a Web form and the results in another. For example, you have the following tag. `<span id="results"></span>` In this case, enter **results** for the property value.
V	ResultXml (see page 457)	Allows a developer to get results from the search control as XML.
	SearchText (see page 457)	Get or set default search text. This is optional.

	ShowCategories (see page 457)	If set to **True**, taxonomy categories will also be searched for the search text specified. The default is False. • **True** = Search (see page 474) across taxonomy categories • **False** = Do not search across taxonomy categories
	ShowCustomSummary (see page 458)	If set to **True**, custom summary is used in search results. The default is False. • **True** = Display custom summary in search results. • **False** = Display characterization generated by indexing service in search results.
	ShowSearchBoxAlways (see page 458)	If set to **False**, the search box does not appear on PostBack. The default is True. • **True** = Display search box on PostBack • **False** = Do not display search box on PostBack
	StartCount (see page 459)	Get the numerical record of the last item on a page.
	Stylesheet (see page 459)	Specify the location of the style sheet for the search results page. Set the location relative to the site root folder. For example: `csslib\mytest.css`. Leave blank to use the default style sheet,
	TaxCategories (see page 459)	Is server control in workarea
	TaxonomyOperator (see page 459)	Select whether to use an **And** or **Or** operator when filtering results by taxonomy. By default, the property is set to **Or**. This allows for a wider range of returned results. • **And** - Only results that match all categories selected in the Filter by Category tree appear. For example, if you are searching for a medical document in the Hospital and Doctor's Office categories, the document must be assigned to both categories or it is not shown. • **Or** - when more than one category is selected in the Filter by Category tree, results must match at least one category... more (see page 459)
	TemplateCart (see page 460)	The URL for the page that contains a basket control (the URL is used when the customer clicks the 'Add to cart' button).
	TemplateProduct (see page 460)	The URL for the page that contains a product control (the URL is used when the customer clicks the image or title link).
	TextBoxSize (see page 461)	The size of the Search (see page 474) text box for user input. The width is based on the number of characters.
	TimeTaken (see page 461)	Gets the amount of time, in seconds, it has taken to perform the search.
	XmlDoc (see page 461)	This property is not supported by WebSearch (see page 529). The underlying XML can be displayed using an XSLT specified by the DisplayXslt (see page 451) property of the WebSearch (see page 529) server control. The XSLT filename is specified relative to the current directory.

ProductSearch.ButtonImgSrc Property

If you want to display an image on the Submit button, enter the server path to it.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("If you want to display an image on the Submit button, enter the server path to
it.")]
public string ButtonImgSrc;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("If you want to display an image on the Submit button, enter the server path to
it.")>
Public Property ButtonImgSrc() As string
```

ProductSearch.ButtonText Property

The text used for the Submit button if no image source is identified. If an image source is identified, this is alternative text for the button.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[Description("The text of the search button if no image is specified. The default value is set
to \"res_srch_btn\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_SEARCH_BTN)]
public string ButtonText;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<Description("The text of the search button if no image is specified. The default value is set
to \"res_srch_btn\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_SEARCH_BTN)>
Public Property ButtonText() As string
```

ProductSearch.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public new double CacheInterval;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public new Property CacheInterval() As double
```

ProductSearch.CatalogId Property

The folder at which the search begins. The starting folder need not be the root folder. The **Recursive (see page 455)** property determines if the search examines this folder's subfolders.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("The catalog at which the search begins. The starting folder need not be the root
folder. The Recursive property determines if the search examines this folder?s subfolders.")]
[Editor(typeof(DefaultCatalogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long CatalogId;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The catalog at which the search begins. The starting folder need not be the root
```

```
folder. The Recursive property determines if the search examines this folder?s subfolders.")>
<Editor(typeof(DefaultCatalogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property CatalogId() As long
```

ProductSearch.CustomOrderBy Property

Provide a property's Friendly Name defined in the Indexing Service to sort search results by that property. Results can be ascending or descending based on `OrderDirection` (see page 455). If you enter an invalid property, no search results are returned. If you specify `CustomOrderBy` and `OrderBy` (see page 454), the `OrderBy` (see page 454) property is ignored. The Friendly Name of a property can be found in **Computer Management** > **Services (see page 1083) and Applications** > **Indexing Service** > **Your Index** > **Properties** > **Friendly Name** column.

C#

```
[Category("Ektron"), DefaultValue(""), Description("Provide a property?s Friendly Name defined
in the Indexing Service to sort search results by that property. Results can be ascending or
descending based on OrderDirection. If you enter an invalid property, no search results are
returned.")]
public string CustomOrderBy;
```

Visual Basic

```
<Category("Ektron"), DefaultValue(""), Description("Provide a property?s Friendly Name defined
in the Indexing Service to sort search results by that property. Results can be ascending or
descending based on OrderDirection. If you enter an invalid property, no search results are
returned.")>
Public Property CustomOrderBy() As string
```

ProductSearch.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses ProductSearch.xsl. This file is located in:

```
[SiteRoot]\Workarea\Xslt\Commerce
```

Important: If you want to edit this file, it is strongly recommended that you make a copy, change its name and move it outside of the Workarea folder. If you make changes to this file and do not move it out of the Workarea, this file and the changes will be lost when you upgrade.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses ProductSearch.xsl.")]
[TypeConverter(typeof(XsltProductSearchTypeConverter))]
[DefaultValue("EcommerceProductSearch")]
[Personalizable(), WebBrowsable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses ProductSearch.xsl.")>
<TypeConverter(typeof(XsltProductSearchTypeConverter))>
<DefaultValue("EcommerceProductSearch")>
<Personalizable(), WebBrowsable()>
Public override Property DisplayXslt() As string
```

ProductSearch.DynamicProductParameter Property

The QueryString parameter name which is used to pass the product ID to the product details page.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The QueryString parameter name which is used to pass the product ID to the
product details page.")]
[DefaultValue("id")]
public string DynamicProductParameter;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The QueryString parameter name which is used to pass the product ID to the
product details page.")>
<DefaultValue("id")>
Public Property DynamicProductParameter() As string
```

ProductSearch.EnableAdvancedLink Property

Set to **True** to display an additional tab (Advanced) on the ProductSearch (see page 445) control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to True to display an additional tab (Advanced) on the ProductSearch
control.")]
public bool EnableAdvancedLink;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to True to display an additional tab (Advanced) on the ProductSearch
control.")>
Public Property EnableAdvancedLink() As bool
```

ProductSearch.EndCount Property

Get the numerical record of the first item on a page.

C#

```
[Browsable(false)]
public int EndCount;
```

Visual Basic

```
<Browsable(false)>
Public Property EndCount() As Integer
```

ProductSearch.IsInWorkArea Property

Is server control in workarea

C#

```
[Browsable(false)]
public bool IsInWorkArea;
```

Visual Basic

```
<Browsable(false)>
Public Property IsInWorkArea() As bool
```

ProductSearch.LinkTarget Property

Gets or sets the target window for the hyperlinks in the search result. The choices are:

- **_blank** - loads the link in a new blank window. This window is not named.
- **_self** - loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.
- **_parent** - loads the link in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.
- **_top** - loads the link in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(ItemLinkTargets._self)]
[Description("The target attribute of the link.")]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(ItemLinkTargets._self)>
<Description("The target attribute of the link.")>
Public Property LinkTarget() As ItemLinkTargets
```

ProductSearch.LoadingImage Property

Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")]
[DefaultValue("ajax-loader_circle_lg.gif")]
public string LoadingImage;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")>
<DefaultValue("ajax-loader_circle_lg.gif")>
Public Property LoadingImage() As string
```

ProductSearch.MaxCharacters Property

The maximum number of characters the Search (see page 474) text box accepts. If you enter a value less than 50, it is a good idea to set the TextBoxSize (see page 461) property to the same number.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(50)]
[Description("The maximum number of characters the Search text box accepts. If you enter a
value less than 50, set the TextBoxSize property to the same number.")]
```

```
public int MaxCharacters;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(50)>
<Description("The maximum number of characters the Search text box accepts. If you enter a
value less than 50, set the TextBoxSize property to the same number.")>
Public Property MaxCharacters() As Integer
```

ProductSearch.MaxTeaserLength Property

Limits the length of any returned content's abstract. To allow unlimited length, set to zero. This property is active only if both of these conditions are true.

- you use the DisplayXslt (see page 451) property to identify an XSLT and ecmteaser as a value of that property.
- the ShowCustomSummary (see page 458) property is set to false. If ShowCustomSummary (see page 458) is set to true, the entire summary appears in search results.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(256)]
[Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")]
public int MaxTeaserLength;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(256)>
<Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")>
Public Property MaxTeaserLength() As Integer
```

ProductSearch.OrderBy Property

The order of the search results. Choices are:

- **Title** - The content title (alphabetical)
- **ID** - The content ID number
- **Date Created** - The date the content was created
- **Date Modified** - The date the content was most recently modified
- **Editor** - The user who last edited the content (alphabetical)
- **Rank** - The rank assigned to the content. See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 725)" > "Display of Search (see page 474) Results" > "Search (see page 474) Result Ranking"

C#

```
[Browsable(true)]
[Category("Ektron")]
[Description("Specify the order of the results. Choices are: Rank, Title, User, DateCreated,
DateModified")]
[Personalizable(), WebBrowsable()]
public WebSearchOrder OrderBy;
```

Visual Basic

```
<Browsable(true)>
```

```
<Category("Ektron")>
<Description("Specify the order of the results. Choices are: Rank, Title, User, DateCreated,
DateModified")>
<Personalizable(), WebBrowsable()>
Public Property OrderBy() As WebSearchOrder
```

ProductSearch.OrderDirection Property

The direction in which search results are sorted. The default is Ascending.

- **Ascending** - Alphabetical results from A to Z; numeric values low to high; dates from oldest to most recent
- **Descending** - Alphabetical results from Z to A; numeric values high to low; dates from most recent to oldest

C#

```
[Browsable(true)]
[Category("Ektron")]
[TypeConverter(typeof(SortOrderTypeConverter))]
[Description("The direction in which search results are sorted. The default is Ascending.
Choices are Ascending and Descending.")]
[Personalizable(), WebBrowsable()]
public string OrderDirection;
```

Visual Basic

```
<Browsable(true)>
<Category("Ektron")>
<TypeConverter(typeof(SortOrderTypeConverter))>
<Description("The direction in which search results are sorted. The default is Ascending.
Choices are Ascending and Descending.")>
<Personalizable(), WebBrowsable()>
Public Property OrderDirection() As string
```

ProductSearch.Recursive Property

Set to True to search sub-folders of the starting folder. The starting folder is identified in the `CatalogID` property.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to True to search sub-folders of the starting folder. The starting folder is
identified in the CatalogID property.")]
public bool Recursive;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to True to search sub-folders of the starting folder. The starting folder is
identified in the CatalogID property.")>
Public Property Recursive() As bool
```

ProductSearch.ResultCount Property

The number of search results. This is a read-only property.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("The number of search results. This is a read-only property.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int ResultCount;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The number of search results. This is a read-only property.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property ResultCount() As Integer
```

ProductSearch.ResultDataTable Property

Allows a developer to get results from the search control as a DataTable

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to get results from the search control as a DataTable.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public virtual DataTable ResultDataTable;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows a developer to get results from the search control as a DataTable.")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Browsable(false)>
Public virtual Property ResultDataTable() As DataTable
```

ProductSearch.ResultsPageSize Property

An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary is truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("An optional page size setting, defaults to web.config search provider page
size.")]
[Personalizable(), WebBrowsable()]
public int ResultsPageSize;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("An optional page size setting, defaults to web.config search provider page
size.")>
<Personalizable(), WebBrowsable()>
Public Property ResultsPageSize() As Integer
```

ProductSearch.ResultTagId Property

Lets you designate where search results appear. You can place search criteria in one area of a Web form and the results in another.

For example, you have the following tag.

```
<span id="results"></span>
```

In this case, enter **results** for the property value.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("an optional parameter to specify the div control id to display the search
result")]
public string ResultTagId;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("an optional parameter to specify the div control id to display the search
result")>
Public Property ResultTagId() As string
```

ProductSearch.ResultXml Property

Allows a developer to get results from the search control as XML.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to get results from the search control as XML.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public virtual string ResultXml;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows a developer to get results from the search control as XML.")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Browsable(false)>
Public virtual Property ResultXml() As string
```

ProductSearch.SearchText Property

Get or set default search text. This is optional.

C#

```
[Browsable(false)]
public string SearchText;
```

Visual Basic

```
<Browsable(false)>
Public Property SearchText() As string
```

ProductSearch.ShowCategories Property

If set to **True**, taxonomy categories will also be searched for the search text specified. The default is False.

- **True** = Search (see page 474) across taxonomy categories

- **False** = Do not search across taxonomy categories

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If this property is set to true, taxonomy categories will also be searched for
the search text specified.")]
public bool ShowCategories;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If this property is set to true, taxonomy categories will also be searched for
the search text specified.")>
Public Property ShowCategories() As bool
```

ProductSearch.ShowCustomSummary Property

If set to **True**, custom summary is used in search results. The default is False.

- **True** = Display custom summary in search results.
- **False** = Display characterization generated by indexing service in search results.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If this property is set to true, custom summary is displayed in search
results.")]
public bool ShowCustomSummary;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If this property is set to true, custom summary is displayed in search
results.")>
Public Property ShowCustomSummary() As bool
```

ProductSearch.ShowSearchBoxAlways Property

If set to **False**, the search box does not appear on PostBack. The default is True.

- **True** = Display search box on PostBack
- **False** = Do not display search box on PostBack

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If set to False, the search box does not appear on PostBack. The default is
True.")]
public bool ShowSearchBoxAlways;
```

Visual Basic

```
<Bindable(true)>
```

```
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If set to False, the search box does not appear on PostBack. The default is
True.")>
Public Property ShowSearchBoxAlways() As bool
```

ProductSearch.StartCount Property

Get the numerical record of the last item on a page.

C#

```
[Browsable(false)]
public int StartCount;
```

Visual Basic

```
<Browsable(false)>
Public Property StartCount() As Integer
```

ProductSearch.Stylesheet Property

Specify the location of the style sheet for the search results page. Set the location relative to the site root folder. For example: `csslib\mytest.css`.

Leave blank to use the default style sheet,

C#

```
[Description("Specify the location of the style sheet for the search results page. Set the
location relative to the site root folder.")]
[Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(ProductSearchCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/ProductSearch.css")]
public string Stylesheet;
```

Visual Basic

```
<Description("Specify the location of the style sheet for the search results page. Set the
location relative to the site root folder.")>
<Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(ProductSearchCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/ProductSearch.css")>
Public Property Stylesheet() As string
```

ProductSearch.TaxCategories Property

Is server control in workarea

C#

```
[Browsable(false)]
public string TaxCategories;
```

Visual Basic

```
<Browsable(false)>
Public Property TaxCategories() As string
```

ProductSearch.TaxonomyOperator Property

Select whether to use an **And** or **Or** operator when filtering results by taxonomy. By default, the property is set to **Or**. This allows for a wider range of returned results.

- **And** - Only results that match all categories selected in the Filter by Category tree appear. For example, if you are searching for a medical document in the Hospital and Doctor's Office categories, the document must be assigned to both categories or it is not shown.
- **Or** - when more than one category is selected in the Filter by Category tree, results must match at least one category to be shown.

Note: For this property to be active, the `ShowCategories` (see page 457) property must be set to **True**.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Taxonomy operator used in filtering search results by taxonomy category.")]
[Personalizable(), WebBrowsable()]
public TaxCategoryOperator TaxonomyOperator;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Taxonomy operator used in filtering search results by taxonomy category.")>
<Personalizable(), WebBrowsable()>
Public Property TaxonomyOperator() As TaxCategoryOperator
```

ProductSearch.TemplateCart Property

The URL for the page that contains a basket control (the URL is used when the customer clicks the 'Add to cart' button).

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The URL for the page that contains a basket control (the URL is used when the
customer clicks the 'Add to cart' button).")]
[DefaultValue("")]
public string TemplateCart;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The URL for the page that contains a basket control (the URL is used when the
customer clicks the 'Add to cart' button).")>
<DefaultValue("")>
Public Property TemplateCart() As string
```

ProductSearch.TemplateProduct Property

The URL for the page that contains a product control (the URL is used when the customer clicks the image or title link).

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The URL for the page that contains a product control (the URL is used when the
customer clicks the image or title link).")]
[DefaultValue("")]
public string TemplateProduct;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The URL for the page that contains a product control (the URL is used when the
customer clicks the image or title link).")>
```

```
<DefaultValue("")>
Public Property TemplateProduct() As string
```

ProductSearch.TextBoxSize Property

The size of the Search (see page 474) text box for user input. The width is based on the number of characters.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The size of the Search text box for user input. The width is based on the number
of characters.")]
public int TextBoxSize;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The size of the Search text box for user input. The width is based on the number
of characters.")>
Public Property TextBoxSize() As Integer
```

ProductSearch.TimeTaken Property

Gets the amount of time, in seconds, it has taken to perform the search.

C#

```
[Browsable(false)]
public double TimeTaken;
```

Visual Basic

```
<Browsable(false)>
Public Property TimeTaken() As double
```

ProductSearch.XmlDoc Property

This property is not supported by WebSearch (see page 529). The underlying XML can be displayed using an XSLT specified by the DisplayXslt (see page 451) property of the WebSearch (see page 529) server control. The XSLT filename is specified relative to the current directory.

C#

```
[Bindable(true)]
[Browsable(false)]
[Description("This property is not supported by WebSearch. To retrieve the XML structure, use
the DisplayXslt property in the WebSearch server control to get search results formatted using
an XSLT. You need to either specify an XSLT name or use the default XSLTs (ecmNavigation,
ecmTeaser) provided with the server control.")]
public override XmlDocument XmlDoc;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Description("This property is not supported by WebSearch. To retrieve the XML structure, use
the DisplayXslt property in the WebSearch server control to get search results formatted using
an XSLT. You need to either specify an XSLT name or use the default XSLTs (ecmNavigation,
ecmTeaser) provided with the server control.")>
Public override Property XmlDoc() As XmlDocument
```

Recommendation Class

Class Hierarchy

C#

```
[DefaultProperty("DefaultProductID"), ToolboxData("<{0}:Recommendation
runat=server></{0}:Recommendation>")]
public class Recommendation : ECommerceControl, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultProductID"), ToolboxData("<{0}:Recommendation
runat=server></{0}:Recommendation>")>
Public Class Recommendation
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

File

Recommendation.cs

Recommendation Methods

The methods of the Recommendation class are listed here.

Public Methods

	Name	Description
	Fill (see page 462)	Populates the data for the Favorites (see page 236).
	ToString (see page 462)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 462)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Recommendation.Fill Method

Populates the data for the Favorites (see page 236).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Recommendation.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Recommendation.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

Recommendation Properties

The properties of the Recommendation class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 463)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DefaultProductID (see page 464)	Enter a default product ID that contains either cross sell or up sell items. To make the server control dynamic, enter zero (0) in this property and set the `DynamicProductParameter` (see page 465) to QueryString parameter used to pass the product's ID.
	DisplayXslt (see page 464)	Specify an external XSLT file to render the control. By default, the control uses `Recommendation.xsl`. This file is located in:
	DynamicCartParameter (see page 465)	The QueryString parameter name which is used to pass the product ID to the Cart (see page 81) server control. For example, if your QueryString parameter for products is Product (see page 431), enter that in this property. Then, when site visitors click the Add to Cart (see page 81) button, this parameter is passed with the product's ID Web form containing the Cart (see page 81) server control.
	DynamicProductParameter (see page 465)	The QueryString parameter name which is used to pass the product ID to the product details page. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors click a product's title, this parameter is passed with the product's ID to the product details page.
	EkItem (see page 466)	Maximum number of results to be returned.
	PageSize (see page 466)	Specify the number of items to show per page when a site visitor views recommendations. If the number of items exceeds the quantity defined in this property, the list is paged and the site visitor can use the paging navigation system to move through the list.
	RecommendationType (see page 466)	The type of recommendation to show. Choices are: • CrossSell - used to help sell additional items that a site visitor might be interested in. For example, if you are selling Denim Jackets, you might want to cross sell them pants or shirts. • UpSell - used to sell additional items that improve on the item being purchased. For example, if you are selling 7 megapixel digital cameras, you might want show the site visitor 10 megapixel digital cameras. For information on Cross Selling and Upselling, see the Administrator Manual section "Adding Cross Sell and Upsell Recommendations to a Catalog Entry"
	Stylesheet (see page 467)	Specify the location of a style sheet to use for the recommendation control. Set the location relative to the site root folder. For example:
	TemplateCart (see page 467)	The URL to navigate to when the site visitor clicks the **Add to Cart (see page 81) button**. This path can be relative or absolute. When a path is entered, the **Add to Cart (see page 81)** button appears next to the item and allows a site visitor to add the item directly to their cart.
	TemplateProduct (see page 467)	Specify the URL that contains the Product (see page 431) server control. This allows a site visitor to see the details of the product when the product link is clicked. This path can be relative or absolute.

Recommendation.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a

control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public new virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public new virtual Property CacheInterval() As double
```

Recommendation.DefaultProductID Property

Enter a default product ID that contains either cross sell or up sell items. To make the server control dynamic, enter zero (0) in this property and set the `DynamicProductParameter` (see page 465) to QueryString parameter used to pass the product's ID.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enter a default product ID that contains either cross sell or up sell items. To
make the server control dynamic, enter zero (0) in this property and set the
DynamicProductParameter to QueryString parameter used to pass the product?s ID.")]
public long DefaultProductID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Enter a default product ID that contains either cross sell or up sell items. To
make the server control dynamic, enter zero (0) in this property and set the
DynamicProductParameter to QueryString parameter used to pass the product?s ID.")>
Public Property DefaultProductID() As long
```

Recommendation.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses `Recommendation.xsl`. This file is located in:

```
[SiteRoot]\Workarea\Xslt\Commerce
```

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses Recommendation.xsl. This file is located in [SiteRoot]\\Workarea\\Xslt\\Commerce")]
[TypeConverter(typeof(XsltRecTypeConverter))]
[DefaultValue("Recommendation.xsl")]
[Personalizable()]
```

```
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses Recommendation.xsl. This file is located in [SiteRoot]\\Workarea\\Xslt\\Commerce")>
<TypeConverter(typeof(XsltRecTypeConverter))>
<DefaultValue("Recommendation.xsl")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

Recommendation.DynamicCartParameter Property

The QueryString parameter name which is used to pass the product ID to the Cart (see page 81) server control. For example, if your QueryString parameter for products is Product (see page 431), enter that in this property. Then, when site visitors click the Add to Cart (see page 81) button, this parameter is passed with the product's ID Web form containing the Cart (see page 81) server control.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The QueryString parameter name which is used to pass the product ID to the Cart
server control. For example, if your QueryString parameter for products is Product, enter that
in this property. Then, when site visitors click the \"Add to Cart\" button, this parameter is
passed with the product?s ID Web form containing the Cart server control.")]
[DefaultValue("product")]
public string DynamicCartParameter;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The QueryString parameter name which is used to pass the product ID to the Cart
server control. For example, if your QueryString parameter for products is Product, enter that
in this property. Then, when site visitors click the \"Add to Cart\" button, this parameter is
passed with the product?s ID Web form containing the Cart server control.")>
<DefaultValue("product")>
Public Property DynamicCartParameter() As string
```

Recommendation.DynamicProductParameter Property

The QueryString parameter name which is used to pass the product ID to the product details page. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors click a product's title, this parameter is passed with the product's ID to the product details page.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("id")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("The QueryString parameter name which is used to pass the product ID to the
product details page. For example, if your QueryString parameter for products is PID, enter
that in this property. Then, when site visitors click a product?s title, this parameter is
passed with the product?s ID to the product details page.")]
[Personalizable()]
public string DynamicProductParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("id")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("The QueryString parameter name which is used to pass the product ID to the
```

```
product details page. For example, if your QueryString parameter for products is PID, enter
that in this property. Then, when site visitors click a product?s title, this parameter is
passed with the product?s ID to the product details page.")>
<Personalizable()>
Public Property DynamicProductParameter() As string
```

Recommendation.EkItem Property

Maximum number of results to be returned.

C#

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Returns Array of Recommendation data.")]
[Personalizable()]
public RecommendationItemData EkItem;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Returns Array of Recommendation data.")>
<Personalizable()>
Public Property EkItem() As RecommendationItemData
```

Recommendation.PageSize Property

Specify the number of items to show per page when a site visitor views recommendations. If the number of items exceeds the quantity defined in this property, the list is paged and the site visitor can use the paging navigation system to move through the list.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("Specify the number of items to show per page when a site visitor views
recommendations. If the number of items exceeds the quantity defined in this property, the
list is paged and the site visitor can use the paging navigation system to move through the
list.")]
[DefaultValue(0)]
public int PageSize;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("Specify the number of items to show per page when a site visitor views
recommendations. If the number of items exceeds the quantity defined in this property, the
list is paged and the site visitor can use the paging navigation system to move through the
list.")>
<DefaultValue(0)>
Public Property PageSize() As Integer
```

Recommendation.RecommendationType Property

The type of recommendation to show. Choices are:

- CrossSell - used to help sell additional items that a site visitor might be interested in. For example, if you are selling Denim Jackets, you might want to cross sell them pants or shirts.
- UpSell - used to sell additional items that improve on the item being purchased. For example, if you are selling 7 megapixel digital cameras, you might want show the site visitor 10 megapixel digital cameras.

For information on Cross Selling and Upselling, see the Administrator Manual section "Adding Cross Sell and Upsell Recommendations to a Catalog Entry"

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(EkEnumeration.RecommendationType.CrossSell),
Description("The type of recommendation to show. Choices are CrossSell and Upsell.")]
public EkEnumeration.RecommendationType RecommendationType;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(EkEnumeration.RecommendationType.CrossSell),
Description("The type of recommendation to show. Choices are CrossSell and Upsell.")>
Public Property RecommendationType() As EkEnumeration.RecommendationType
```

Recommendation.Stylesheet Property

Specify the location of a style sheet to use for the recommendation control. Set the location relative to the site root folder. For example:

```
Workarea\csslib\mytest.css
```

Leave blank to use the default style sheet.

C#

```
[Category("Ektron")]
[Description("Specify the location of a style sheet to use for the recomendation. Set the
location relative to the site root folder.")]
[TypeConverter(typeof(RecommendationCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/Recommendation.css")]
public string Stylesheet;
```

Visual Basic

```
<Category("Ektron")>
<Description("Specify the location of a style sheet to use for the recomendation. Set the
location relative to the site root folder.")>
<TypeConverter(typeof(RecommendationCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/Recommendation.css")>
Public Property Stylesheet() As string
```

Recommendation.TemplateCart Property

The URL to navigate to when the site visitor clicks the **Add to Cart (see page 81) button**. This path can be relative or absolute. When a path is entered, the **Add to Cart (see page 81)** button appears next to the item and allows a site visitor to add the item directly to their cart.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The URL to navigate to when the site visitor clicks the \"Add to Cart\" button.
This path can be relative or absolute. When a path is entered, the \"Add to Cart\" button
appears next to the item and allows a site visitor to add the item directly to their cart.")]
[DefaultValue("")]
public string TemplateCart;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The URL to navigate to when the site visitor clicks the \"Add to Cart\" button.
This path can be relative or absolute. When a path is entered, the \"Add to Cart\" button
appears next to the item and allows a site visitor to add the item directly to their cart.")>
<DefaultValue("")>
Public Property TemplateCart() As string
```

Recommendation.TemplateProduct Property

Specify the URL that contains the Product (see page 431) server control. This allows a site visitor to see the details of the

product when the product link is clicked. This path can be relative or absolute.

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("Specify the URL that contains the Product server control. This allows a site
visitor to see the details of the product when the product link is clicked. This path can be
relative or absolute.")]
[DefaultValue("")]
public string TemplateProduct;
```

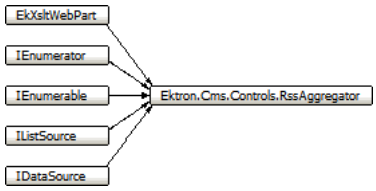
Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("Specify the URL that contains the Product server control. This allows a site
visitor to see the details of the product when the product link is clicked. This path can be
relative or absolute.")>
<DefaultValue("")>
Public Property TemplateProduct() As string
```

RssAggregator Class

Ektron RssAggregator class. Allows access to a RSS Feed object through IEnumerator, ICollection and IList interfaces.

Class Hierarchy



C#

```
public class RssAggregator : EkXsltWebPart, IEnumerator, IEnumerable, IListSource, IDataSource;
```

Visual Basic

```
Public Class RssAggregator
Inherits EkXsltWebPart
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
```

File

RssAggregator.cs

RssAggregator Methods

The methods of the RssAggregator class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 469)	Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.
	Fill (see page 469)	Calls up the database and updates the contents of the collection if the data is out of sync with the specified parameters.
	GetEnumerator (see page 469)	Gets a System.Collections.IEnumerator for the collection.
	GetView (see page 469)	Gets the view data using dataset

	GetViewNames (see page 469)	gets the view names
	MoveNext (see page 470)	Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.
	Reset (see page 470)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 470)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.

RssAggregator.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

C#

```
public void CopyTo(Array array, int i);
```

Visual Basic

```
Public Function CopyTo(array As Array, i As Integer) As void
```

RssAggregator.Fill Method

Calls up the database and updates the contents of the collection if the data is out of sync with the specified parameters.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

RssAggregator.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

```
public IEnumerator GetEnumerator();
```

Visual Basic

```
Public Function GetEnumerator() As IEnumerator
```

RssAggregator.GetView Method

Gets the view data using dataset

C#

```
public DataSourceView GetView(string viewName);
```

Visual Basic

```
Public Function GetView(viewName As string) As DataSourceView
```

Parameters

Parameters	Description
string viewName	

Returns

DataSourceView

RssAggregator.GetViewNames Method

gets the view names

C#

```
public ICollection GetViewNames();
```

Visual Basic

```
Public Function GetViewNames() As ICollection
```

Returns

View names

RssAggregator.MoveNext Method

Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.

C#

```
public bool MoveNext();
```

Visual Basic

```
Public Function MoveNext() As bool
```

Returns

Boolean indicating if the move was within the range of the array.

RssAggregator.Reset Method

Brings the index of the collection back to zero. Does not clear the collection.

C#

```
public void Reset();
```

Visual Basic

```
Public Function Reset() As void
```

RssAggregator.this Indexer

Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.

C#

```
public Ektron.Cms.Controls.RssItem._ItemRow this[int i];
```

Visual Basic

```
Public Sub this(i As Integer)
```

RssAggregator Properties

The properties of the RssAggregator class are listed here.

Public Properties

	Name	Description
	Count (see page 471)	Indicates the number of elements in your Collection (see page 95).
	Current (see page 471)	Gets the current object in the collection.
	EkItems (see page 471)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 95). Holds more information than the exposed strongly typed this[i] index operator.
	IsSynchronized (see page 471)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

	LinkTarget (see page 472)	Gets or Sets the target window for the result. Choices are: • **_blank** - This target causes the link to always be loaded in a new blank window. This window is not named. • **_self** - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target. • **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** - This target makes the link load in the full... more (see page 472)
	MaxResults (see page 472)	The maximum number of results to fetch. If set to zero, there is no limit.
	SyncRoot (see page 472)	Gets an object that can be used to synchronize the collection.
	URL (see page 473)	Gets or Sets the RSS Feed URL for this Control. The RSS feed path for the server control. For example: http://msdn.microsoft.com/rss.xml
	WrapTag (see page 473)	Allows a developer to specify control's tag. Default is assumed "".

RssAggregator.Count Property

Indicates the number of elements in your Collection (see page 95).

C#

```
[Category("Data")]
[Description("Indicates the number of elements in your Collection.")]
public int Count;
```

Visual Basic

```
<Category("Data")>
<Description("Indicates the number of elements in your Collection.")>
Public Property Count() As Integer
```

RssAggregator.Current Property

Gets the current object in the collection.

C#

```
[Browsable(false)]
public object Current;
```

Visual Basic

```
<Browsable(false)>
Public Property Current() As object
```

RssAggregator.EkItems Property

Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 95). Holds more information than the exposed strongly typed this[i] index operator.

C#

```
[Browsable(false)]
public RssItemBase EkItems;
```

Visual Basic

```
<Browsable(false)>
Public Property EkItems() As RssItemBase
```

RssAggregator.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

```
[Browsable(false)]
public bool IsSynchronized;
```

Visual Basic

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

RssAggregator.LinkTarget Property

Gets or Sets the target window for the result. Choices are:

- **_blank** - This target causes the link to always be loaded in a new blank window. This window is not named.
- **_self** - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.
- **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.
- **_top** - This target makes the link load in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
[Personalizable(), WebBrowsable()]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
<Personalizable(), WebBrowsable()>
Public Property LinkTarget() As ItemLinkTargets
```

RssAggregator.MaxResults Property

The maximum number of results to fetch. If set to zero, there is no limit.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The maximum number of results to fetch. If set to zero, there is no limit.")]
[Personalizable(), WebBrowsable()]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The maximum number of results to fetch. If set to zero, there is no limit.")>
<Personalizable(), WebBrowsable()>
Public Property MaxResults() As Integer
```

RssAggregator.SyncRoot Property

Gets an object that can be used to synchronize the collection.

C#

```
[Browsable(false)]
public object SyncRoot;
```

Visual Basic

```
<Browsable(false)>
Public Property SyncRoot() As object
```

RssAggregator.URL Property

Gets or Sets the RSS Feed URL for this Control. The RSS feed path for the server control. For example: http://msdn.microsoft.com/rss.xml

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Gets or Sets the RSS Feed URL for this Control.")]
[Personalizable(), WebBrowsable()]
public string URL;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Gets or Sets the RSS Feed URL for this Control.")>
<Personalizable(), WebBrowsable()>
Public Property URL() As string
```

RssAggregator.WrapTag Property

Allows a developer to specify control's tag. Default is assumed "".

C#

```
[Browsable(false)]
public override string WrapTag;
```

Visual Basic

```
<Browsable(false)>
Public override Property WrapTag() As string
```

RssItemBase Class

Container for Title (see page 474), Link (see page 474), Description (see page 474) and DatePosted (see page 474) article information.

Class Hierarchy

Ektron.Cms.Controls.RssItemBase

C#

```
public class RssItemBase;
```

Visual Basic

```
Public Class RssItemBase
```

File

RssAggregator.cs

RssItemBase Fields

The fields of the RssItemBase class are listed here.

Public Fields

	Name	Description
	DatePosted (see page 474)	The date the article was posted.
	Description (see page 474)	The description of the article.
	Link (see page 474)	The link for the article.
	Title (see page 474)	The title of the article.

RssItemBase.DatePosted Field

The date the article was posted.

C#

```
public DateTime DatePosted = new DateTime(0);
```

Visual Basic

```
Public DatePosted As DateTime = new DateTime(0)
```

RssItemBase.Description Field

The description of the article.

C#

```
public string Description = "";
```

Visual Basic

```
Public Description As string = ""
```

RssItemBase.Link Field

The link for the article.

C#

```
public string Link = "";
```

Visual Basic

```
Public Link As string = ""
```

RssItemBase.Title Field

The title of the article.

C#

```
public string Title = "";
```

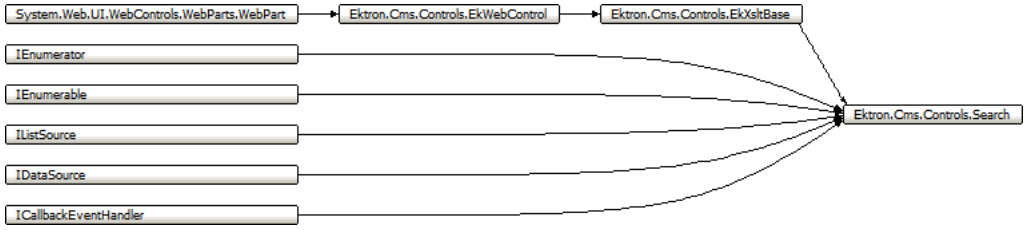
Visual Basic

```
Public Title As string = ""
```

Search Class

Search object returns the control and results.

Class Hierarchy



C#

```
[DefaultProperty("FolderID"), ToolboxData("<{0}:Search runat=server></{0}:Search>")]
[Obsolete()]
[ToolboxItemFilter("Ektron.Cms.Controls.Search", ToolboxItemFilterType.Prevent)]
public class Search : EkXsltBase, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("FolderID"), ToolboxData("<{0}:Search runat=server></{0}:Search>")>
<Obsolete()>
<ToolboxItemFilter("Ektron.Cms.Controls.Search", ToolboxItemFilterType.Prevent)>
Public Class Search
Inherits EkXsltBase
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
Inherits ICallbackEventHandler
```

File

Search.cs

Search.Search Constructor

Constructor for Search class

C#

```
public Search();
```

Visual Basic

```
Public Sub New()
```

Search Methods

The methods of the Search class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 476)	Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.
	Fill (see page 476)	Method that can be overridden and used to grab data from the back end for each server control.
	GetEnumerator (see page 476)	Gets a System.Collections.IEnumerator for the collection.
	LoadSearch (see page 476)	
	MoveNext (see page 476)	Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.
	Reset (see page 477)	Brings the index of the collection back to zero. Does not clear the collection.

	SearchDisplayUI (see page 477)	
	this (see page 477)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.
	Wrap (see page 477)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Search.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

C#

```
public void CopyTo(Array array, int index);
```

Visual Basic

```
Public Function CopyTo(array As Array, index As Integer) As void
```

Search.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Search.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

```
public IEnumerator GetEnumerator();
```

Visual Basic

```
Public Function GetEnumerator() As IEnumerator
```

Search.LoadSearch Method

C#

```
public void LoadSearch(ekSearchConfig SearchReq);
```

Visual Basic

```
Public Function LoadSearch(SearchReq As ekSearchConfig) As void
```

Parameters

Parameters	Description
ekSearchConfig SearchReq	

Search.MoveNext Method

Moves up the index of the Collection (see page 95). Returns false if the end of the collection has been surpassed.

C#

```
public bool MoveNext();
```

Visual Basic

```
Public Function MoveNext() As bool
```

Returns

Boolean indicating if the move was within the range of the array.

Search.Reset Method

Brings the index of the collection back to zero. Does not clear the collection.

C#

```
public void Reset();
```

Visual Basic

```
Public Function Reset() As void
```

Search.SearchDisplayUI Method

C#

```
public virtual string SearchDisplayUI(bool supressPostRedirect);
```

Visual Basic

```
Public virtual Function SearchDisplayUI(supressPostRedirect As bool) As string
```

Parameters

Parameters	Description
bool supressPostRedirect	

Search.this Indexer

Index operator into a strongly typed dataset holding some common properties for a Collection (see page 95) item.

C#

```
public Ektron.Cms.Controls.ContentItem.ContentRow this[int i];
```

Visual Basic

```
Public Sub this(i As Integer)
```

Search.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

Search Properties

The properties of the Search class are listed here.

Public Properties

	Name	Description
	ButtonImgSrc (see page 480)	Optional: The path to the image you wish to use for the button.
	ButtonText (see page 480)	The text of the search button if no image is specified.
	ContentFieldsetLegend (see page 480)	Text (see page 228) that appears in the fieldset border around the content check boxes. This property is used only when the EnableContentFieldset (see page 484) property is set to True. The default is: **Content (see page 822) Items:**
	Count (see page 481)	Indicates the number of elements in the search results.
	Current (see page 481)	Gets the current object in the collection.
	Display (see page 481)	Determines if the search options are in a row (**Horizontal**) or a column (**Vertical**).
	EkItems (see page 481)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 95). Holds more information than the exposed strongly typed this[i] index operator.
	EmptyResultMsg (see page 482)	Optional: The return massage if the search returns no hits.
	EnableAdvancedLink (see page 482)	Set to true to display the link that makes the advanced properties visible.
	EnableAjax (see page 482)	By default, the search is set to PostBack. To enable an Ajax search, set this property to true. Ajax provides a quicker response time because the client only loads the results not the whole page.
	EnableArchived (see page 483)	Set to true to include Archived items in the search results.
	EnableArchivedBtn (see page 483)	Set to true to provide a check box so users can decide if they want to perform a Archived search. When set to False the check box is hidden but still used.
	EnableContent (see page 483)	Set to true to include Content (see page 822) items in the search results.
	EnableContentBtn (see page 483)	Set to true to provide a check box so users can decide if they want to perform a content search. When set to False the check box is hidden but still used.
	EnableContentFieldset (see page 484)	Creates a border around the group of content items in an advanced search. These content items include Content (see page 822), Forms and DMS Assets. The default is False.
	EnableDMSAssets (see page 484)	Set to true to include DMS-Asset (see page 822) items in the search results.
	EnableDMSAssetsBtn (see page 484)	Set to true to provide a check box so users can decide if they want to perform a DMS-Asset (see page 822) search. When set to False the check box is hidden but still used.
	EnableFolderFieldset (see page 485)	Set to true to wrap the Folder (see page 683)-Descriptions in a fieldset border.
	EnableForms (see page 485)	Set to true to include Form (see page 871) items in the search results.
	EnableFormsBtn (see page 485)	Set to true to provide a check box so users can decide if they want to perform a Forms search. When set to False the check box is hidden but still used.
	EnableLibFiles (see page 486)	Set to true to include Library (see page 695)-File items in the search results.
	EnableLibFilesBtn (see page 486)	Set to true to provide a check box so users can decide if they want to perform a Library (see page 695)-Files search. When set to False the check box is hidden but still used.
	EnableLibHyperlinks (see page 486)	Set to true to include Library (see page 695)-Hyperlink items in the search results.
	EnableLibHyperlinksBtn (see page 486)	Set to true to provide a check box so users can decide if they want to perform a Library (see page 695)-Hyperlinks search. When set to False the check box is hidden but still used.
	EnableLibImages (see page 487)	Set to true to include Library (see page 695)-Image items in the search results.
	EnableLibImagesBtn (see page 487)	Set to true to display the Library (see page 695)-Images check box (false to hide, but still use).
	EnableLibraryFieldset (see page 487)	Set to true to wrap the Library (see page 695)-Group in a fieldset border.
	FolderFieldsetLegend (see page 488)	The Legend-text to display, for the Folder (see page 683)-Descriptions fieldset.
	FolderID (see page 488)	The folder at which the search begins. The starting folder need not be the root folder. The Recursive (see page 492) property determines if the search examines this folder's subfolders.
	FolderPath (see page 488)	The Path of the folder you wish to start the search. Root starts at "\".
	IncludeIcons (see page 489)	Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary AND ecmTeaser or or when the [$imageIcon] variable is used in an EkML file.
	IsSynchronized (see page 489)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

	LibraryFieldsetLegend (see page 489)	The Legend-text to display, for the Library (see page 695)-Group in a fieldset.
	LinkTarget (see page 490)	Gets or Sets the target window for the result. The choices are: • **_blank** - this target causes the link to always be loaded in a new blank window. This window is not named. • **_self** - this target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target. • **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** - This target makes the link load in the... more (see page 490)
	MaxCharacters (see page 490)	The maximum number of characters the user can enter in the search box.
	MaxResults (see page 490)	The total number of results to display. If set to zero, unlimited results will be set.
	MaxTeaserLength (see page 491)	An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary will be truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT).
	OrderBy (see page 491)	Specify the order of the results. Choices are: • **Title** - the title of the content • **Id** - the ID number content • **DateCreated** - the date the creation date of the content • **DateModified** - the date the content was last modified • **UserId** - ID of a user • **LastEditorLname** - the last name of the last content editor • **LastEditorFname** - the first name of the last content editor
	OrderDirection (see page 491)	Specify the order direction of the results. Choices are Ascending or Descending.
	PostResultsAt (see page 492)	The Path of the page you wish to post search results. (NOTE: The search control MUST exists on the posting page.)
	Recursive (see page 492)	Get or set whether to search subfolders of the identified root folder. The root folder is identified in the FolderID (see page 488) property.
	RemoveTeaserHtml (see page 492)	An optional result formatting parameter. If true, any HTML in the content items' teaser/summary is removed before it is sent to the browser. If false then any existing HTML is passed along, (this may then be further filtered by the XSLT).
	ResultTagId (see page 493)	Specify the div tag control ID to display the search results. This property is only active when the EnableAjax (see page 482) property is set to true.
	SearchText (see page 493)	Get or set the default search text.
	SearchType (see page 493)	Get or set the Search (see page 474) Type
	ShowDate (see page 493)	Select true if you wish to display the publication date on each search result.
	ShowExtendedSearch (see page 494)	Allows a user to search via Searchable Property metadata. • **True** = Show extended search • **False** = Hide (see page 226) extended search Search (see page 474) type metadata only appears on the search screen if the following are true: • this property is set to True • the ShowSearchOptions (see page 494) property is set to True • at least one content item in the starting folder (and its subfolders if recursive is set to true) has searchable metadata applied to it

	ShowSearchBoxAlways (see page 494)	If set to false, the search box does not appear on PostBack. The default is True. • **True** = display search box on PostBack • **False** = do not display search box on PostBack
	ShowSearchOptions (see page 494)	Specify "True" to show search options. Set to false to see a very basic UI.
	Stylesheet (see page 495)	Specify the style sheet relative to site root. Leave blank for default.
	SyncRoot (see page 495)	Gets an object that can be used to synchronize the collection.
	TargetFolderDescriptionList (see page 495)	An optional comma-delimited string, which allows the user to override the folders to search. For example, set to "Photos=1,NewsHeadlines=0" to search only the Photos folder by default, and allow the user to also select the "NewsHeadlines" folder. Also searches these folders' subfolders, returning any matches found there.
	TextBoxSize (see page 496)	The width of the search box, in characters.
	XmlDoc (see page 496)	Returns the Xslt object for the collection.

Search.ButtonImgSrc Property

Optional: The path to the image you wish to use for the button.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Optional: The path to the image you wish to use for the button.")]
public string ButtonImgSrc;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Optional: The path to the image you wish to use for the button.")>
Public Property ButtonImgSrc() As string
```

Search.ButtonText Property

The text of the search button if no image is specified.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The text of the search button if no image is specified. The default value is set
to \"res_srch_btn\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_SEARCH_BTN)]
public string ButtonText;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The text of the search button if no image is specified. The default value is set
to \"res_srch_btn\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_SEARCH_BTN)>
Public Property ButtonText() As string
```

Search.ContentFieldsetLegend Property

Text (see page 228) that appears in the fieldset border around the content check boxes. This property is used only when the EnableContentFieldset (see page 484) property is set to True. The default is: **Content (see page 822) Items:**

C#

```
[Bindable(true)]
```

```
[Category("Ektron")]
[DefaultValue("Content Items: ")]
[Description("The Legend-text to display, for the Content-Group fieldset.")]
public string ContentFieldsetLegend;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Content Items: ")>
<Description("The Legend-text to display, for the Content-Group fieldset.")>
Public Property ContentFieldsetLegend() As string
```

Search.Count Property

Indicates the number of elements in the search results.

C#

```
[Category("Data")]
[Description("Indicates the number of elements in the search results.")]
public int Count;
```

Visual Basic

```
<Category("Data")>
<Description("Indicates the number of elements in the search results.")>
Public Property Count() As Integer
```

Search.Current Property

Gets the current object in the collection.

C#

```
[Browsable(false)]
public object Current;
```

Visual Basic

```
<Browsable(false)>
Public Property Current() As object
```

Search.Display Property

Determines if the search options are in a row (**Horizontal**) or a column (**Vertical**).

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The display format of the image - horizontal or vertical.")]
public DisplayFormat Display;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The display format of the image - horizontal or vertical.")>
Public Property Display() As DisplayFormat
```

Search.EkItems Property

Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 95). Holds more information than the exposed strongly typed this[i] index operator.

C#

```
public Ektron.Cms.Common.SearchContentItem EkItems;
```

Visual Basic

```
Public Property EkItems() As Ektron.Cms.Common.SearchContentItem
```

Search.EmptyResultMsg Property

Optional: The return massage if the search returns no hits.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Optional: The return message if the search returns no hits. The default value is
set to \"res_srch_noHits\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_SEARCH_NOHITS)]
public string EmptyResultMsg;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Optional: The return message if the search returns no hits. The default value is
set to \"res_srch_noHits\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_SEARCH_NOHITS)>
Public Property EmptyResultMsg() As string
```

Search.EnableAdvancedLink Property

Set to true to display the link that makes the advanced properties visible.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the link that makes the advanced properties visible.")]
public bool EnableAdvancedLink;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the link that makes the advanced properties visible.")>
Public Property EnableAdvancedLink() As bool
```

Search.EnableAjax Property

By default, the search is set to PostBack. To enable an Ajax search, set this property to true. Ajax provides a quicker response time because the client only loads the results not the whole page.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Enables Ajax type of search. Ajax provides a quicker response time because the
client only loads the results not the whole page.")]
public bool EnableAjax;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
```

```
<DefaultValue(true)>
<Description("Enables Ajax type of search. Ajax provides a quicker response time because the
client only loads the results not the whole page.")>
Public Property EnableAjax() As bool
```

Search.EnableArchived Property

Set to true to include Archived items in the search results.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to true to include Archived items in the search results.")]
public bool EnableArchived;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to include Archived items in the search results.")>
Public Property EnableArchived() As bool
```

Search.EnableArchivedBtn Property

Set to true to provide a check box so users can decide if they want to perform a Archived search. When set to False the check box is hidden but still used.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to true to display the Archived check box (false to hide, but still use).")]
public bool EnableArchivedBtn;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to display the Archived check box (false to hide, but still use).")>
Public Property EnableArchivedBtn() As bool
```

Search.EnableContent Property

Set to true to include Content (see page 822) items in the search results.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to include Content items in the search results.")]
public bool EnableContent;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to include Content items in the search results.")>
Public Property EnableContent() As bool
```

Search.EnableContentBtn Property

Set to true to provide a check box so users can decide if they want to perform a content search. When set to False the check box

is hidden but still used.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the Content check box (false to hide, but still use).")]
public bool EnableContentBtn;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the Content check box (false to hide, but still use).")>
Public Property EnableContentBtn() As bool
```

Search.EnableContentFieldset Property

Creates a border around the group of content items in an advanced search. These content items include Content (see page 822), Forms and DMS Assets. The default is False.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to true to wrap the Content-Group in a fieldset border.")]
public bool EnableContentFieldset;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to wrap the Content-Group in a fieldset border.")>
Public Property EnableContentFieldset() As bool
```

Search.EnableDMSAssets Property

Set to true to include DMS-Asset (see page 822) items in the search results.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to include DMS-Asset items in the search results.")]
public bool EnableDMSAssets;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to include DMS-Asset items in the search results.")>
Public Property EnableDMSAssets() As bool
```

Search.EnableDMSAssetsBtn Property

Set to true to provide a check box so users can decide if they want to perform a DMS-Asset (see page 822) search. When set to False the check box is hidden but still used.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
```

```
[Description("Set to true to display the DMS-Asset check box (false to hide, but still use).")]
public bool EnableDMSAssetsBtn;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the DMS-Asset check box (false to hide, but still use).")>
Public Property EnableDMSAssetsBtn() As bool
```

Search.EnableFolderFieldset Property

Set to true to wrap the Folder (see page 683)-Descriptions in a fieldset border.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to true to wrap the Folder-Descriptions in a fieldset border.")]
public bool EnableFolderFieldset;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to wrap the Folder-Descriptions in a fieldset border.")>
Public Property EnableFolderFieldset() As bool
```

Search.EnableForms Property

Set to true to include Form (see page 871) items in the search results.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to include Form items in the search results.")]
public bool EnableForms;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to include Form items in the search results.")>
Public Property EnableForms() As bool
```

Search.EnableFormsBtn Property

Set to true to provide a check box so users can decide if they want to perform a Forms search. When set to False the check box is hidden but still used.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the Forms check box (false to hide, but still use).")]
public bool EnableFormsBtn;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the Forms check box (false to hide, but still use).")>
```

```
Public Property EnableFormsBtn() As bool
```

Search.EnableLibFiles Property

Set to true to include Library (see page 695)-File items in the search results.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to include Library-File items in the search results.")]
public bool EnableLibFiles;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to include Library-File items in the search results.")>
Public Property EnableLibFiles() As bool
```

Search.EnableLibFilesBtn Property

Set to true to provide a check box so users can decide if they want to perform a Library (see page 695)-Files search. When set to False the check box is hidden but still used.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the Library-Files check box (false to hide, but still
use).")]
public bool EnableLibFilesBtn;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the Library-Files check box (false to hide, but still
use).")>
Public Property EnableLibFilesBtn() As bool
```

Search.EnableLibHyperlinks Property

Set to true to include Library (see page 695)-Hyperlink items in the search results.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to include Library-Hyperlink items in the search results.")]
public bool EnableLibHyperlinks;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to include Library-Hyperlink items in the search results.")>
Public Property EnableLibHyperlinks() As bool
```

Search.EnableLibHyperlinksBtn Property

Set to true to provide a check box so users can decide if they want to perform a Library (see page 695)-Hyperlinks search. When set to False the check box is hidden but still used.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the Library-Hyperlinks check box (false to hide, but
still use).")]
public bool EnableLibHyperlinksBtn;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the Library-Hyperlinks check box (false to hide, but
still use).")>
Public Property EnableLibHyperlinksBtn() As bool
```

Search.EnableLibImages Property

Set to true to include Library (see page 695)-Image items in the search results.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to include Library-Image items in the search results.")]
public bool EnableLibImages;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to include Library-Image items in the search results.")>
Public Property EnableLibImages() As bool
```

Search.EnableLibImagesBtn Property

Set to true to display the Library (see page 695)-Images check box (false to hide, but still use).

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the Library-Images check box (false to hide, but still
use).")]
public bool EnableLibImagesBtn;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the Library-Images check box (false to hide, but still
use).")>
Public Property EnableLibImagesBtn() As bool
```

Search.EnableLibraryFieldset Property

Set to true to wrap the Library (see page 695)-Group in a fieldset border.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
```

```
[Description("Set to true to wrap the Library-Group in a fieldset border.")]
public bool EnableLibraryFieldset;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to wrap the Library-Group in a fieldset border.")>
Public Property EnableLibraryFieldset() As bool
```

Search.FolderFieldsetLegend Property

The Legend-text to display, for the Folder (see page 683)-Descriptions fieldset.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Folder Descriptions: ")]
[Description("The Legend-text to display, for the Folder-Descriptions fieldset.")]
public string FolderFieldsetLegend;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Folder Descriptions: ")>
<Description("The Legend-text to display, for the Folder-Descriptions fieldset.")>
Public Property FolderFieldsetLegend() As string
```

Search.FolderID Property

The folder at which the search begins. The starting folder need not be the root folder. The Recursive (see page 492) property determines if the search examines this folder's subfolders.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The folder at which the search begins. The starting folder need not be the root
folder. The Recursive property determines if the search examines this folder?s subfolders.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The folder at which the search begins. The starting folder need not be the root
folder. The Recursive property determines if the search examines this folder?s subfolders.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderID() As long
```

Search.FolderPath Property

The Path of the folder you wish to start the search. Root starts at "\".

C#

```
[Bindable(true)]
[Category("Ektron")]
[Browsable(false)]
[DefaultValue("")]
[Description("The Path of the folder you wish to start the search. Root starts at \"\\\".")]
```

```
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public string FolderPath;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Browsable(false)>
<DefaultValue("")>
<Description("The Path of the folder you wish to start the search. Root starts at \"\\\".")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderPath() As string
```

Search.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary AND ecmTeaser or or when the [$imageIcon] variable is used in an EkML file.

C#

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
public bool IncludeIcons;
```

Visual Basic

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")>
<DefaultValue(false)>
<Category("Ektron")>
Public Property IncludeIcons() As bool
```

Search.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

```
[Browsable(false)]
public bool IsSynchronized;
```

Visual Basic

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

Search.LibraryFieldsetLegend Property

The Legend-text to display, for the Library (see page 695)-Group in a fieldset.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Library Items: ")]
[Description("The Legend-text to display, for the Library-Group fieldset.")]
public string LibraryFieldsetLegend;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Library Items: ")>
<Description("The Legend-text to display, for the Library-Group fieldset.")>
```

```
Public Property LibraryFieldsetLegend() As string
```

Search.LinkTarget Property

Gets or Sets the target window for the result. The choices are:

- **_blank** - this target causes the link to always be loaded in a new blank window. This window is not named.
- **_self** - this target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.
- **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.
- **_top** - This target makes the link load in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

Search.MaxCharacters Property

The maximum number of characters the user can enter in the search box.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(50)]
[Description("The maximum number of characters the user can enter in the search box.")]
public int MaxCharacters;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(50)>
<Description("The maximum number of characters the user can enter in the search box.")>
Public Property MaxCharacters() As Integer
```

Search.MaxResults Property

The total number of results to display. If set to zero, unlimited results will be set.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The total number of results to display. If set to zero, unlimited results will
be set.")]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The total number of results to display. If set to zero, unlimited results will
be set.")>
Public Property MaxResults() As Integer
```

Search.MaxTeaserLength Property

An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary will be truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT).

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(100)]
[Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")]
public int MaxTeaserLength;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(100)>
<Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")>
Public Property MaxTeaserLength() As Integer
```

Search.OrderBy Property

Specify the order of the results. Choices are:

- **Title** - the title of the content
- **Id** - the ID number content
- **DateCreated** - the date the creation date of the content
- **DateModified** - the date the content was last modified
- **UserId** - ID of a user
- **LastEditorLname** - the last name of the last content editor
- **LastEditorFname** - the first name of the last content editor

C#

```
[Browsable(true), Category("Ektron"), Description("Specify the order of the results. Choices
are: Title, Id, DateCreated, DateModified, UserId, LastEditorLname, LastEditorFname.")]
public Ektron.Cms.Controls.SearchOrder OrderBy;
```

Visual Basic

```
<Browsable(true), Category("Ektron"), Description("Specify the order of the results. Choices
are: Title, Id, DateCreated, DateModified, UserId, LastEditorLname, LastEditorFname.")>
Public Property OrderBy() As Ektron.Cms.Controls.SearchOrder
```

Search.OrderDirection Property

Specify the order direction of the results. Choices are Ascending or Descending.

C#

```
[Category("Ektron"), TypeConverter(typeof(SortOrderTypeConverter)), Description("Specify the
order direction of the results.")]
public string OrderDirection;
```

Visual Basic

```
<Category("Ektron"), TypeConverter(typeof(SortOrderTypeConverter)), Description("Specify the
order direction of the results.")>
Public Property OrderDirection() As string
```

Search.PostResultsAt Property

The Path of the page you wish to post search results. (NOTE: The search control MUST exists on the posting page.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("#")]
[Description("The Path of the page you wish to post search results. (NOTE: The search control
MUST exists on the posting page.) ")]
public string PostResultsAt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("#")>
<Description("The Path of the page you wish to post search results. (NOTE: The search control
MUST exists on the posting page.) ")>
Public Property PostResultsAt() As string
```

Search.Recursive Property

Get or set whether to search subfolders of the identified root folder. The root folder is identified in the FolderID (see page 488) property.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Select true if you wish to search the subfolders of the selected Folder")]
public bool Recursive;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Select true if you wish to search the subfolders of the selected Folder")>
Public Property Recursive() As bool
```

Search.RemoveTeaserHtml Property

An optional result formatting parameter. If true, any HTML in the content items' teaser/summary is removed before it is sent to the browser. If false then any existing HTML is passed along, (this may then be further filtered by the XSLT).

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("An optional search-result formatting parameter that removes all HTML from each
content-items' teaser/summary before sending it to the browser.")]
public bool RemoveTeaserHtml;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("An optional search-result formatting parameter that removes all HTML from each
content-items' teaser/summary before sending it to the browser.")>
Public Property RemoveTeaserHtml() As bool
```

Search.ResultTagId Property

Specify the div tag control ID to display the search results. This property is only active when the EnableAjax (see page 482) property is set to true.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("An optional parameter to specify the div control id to display the search
result.")]
public string ResultTagId;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("An optional parameter to specify the div control id to display the search
result.")>
Public Property ResultTagId() As string
```

Search.SearchText Property

Get or set the default search text.

C#

```
[Browsable(false)]
public string SearchText;
```

Visual Basic

```
<Browsable(false)>
Public Property SearchText() As string
```

Search.SearchType Property

Get or set the Search (see page 474) Type

C#

```
[Browsable(false)]
public Ektron.Cms.Controls.SearchTypes SearchType;
```

Visual Basic

```
<Browsable(false)>
Public Property SearchType() As Ektron.Cms.Controls.SearchTypes
```

Search.ShowDate Property

Select true if you wish to display the publication date on each search result.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Select true if you wish to display the publication date on each search result.")]
public bool ShowDate;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Select true if you wish to display the publication date on each search result.")>
Public Property ShowDate() As bool
```

Search.ShowExtendedSearch Property

Allows a user to search via Searchable Property metadata.

- **True** = Show extended search
- **False** = Hide (see page 226) extended search

Search (see page 474) type metadata only appears on the search screen if the following are true:

- this property is set to True
- the ShowSearchOptions (see page 494) property is set to True
- at least one content item in the starting folder (and its subfolders if recursive is set to true) has searchable metadata applied to it

C#

```
[Category("Ektron"), Description("Set to true to allow a user to search via the metadata type:
Searchable Property.")]
public bool ShowExtendedSearch;
```

Visual Basic

```
<Category("Ektron"), Description("Set to true to allow a user to search via the metadata type:
Searchable Property.")>
Public Property ShowExtendedSearch() As bool
```

Notes

The folderID property determines the search's starting folder.

Search.ShowSearchBoxAlways Property

If set to false, the search box does not appear on PostBack. The default is True.

- **True** = display search box on PostBack
- **False** = do not display search box on PostBack

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to false, the search box will not be visible on
PostBack.")]
public bool ShowSearchBoxAlways;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to false, the search box will not be visible on
PostBack.")>
Public Property ShowSearchBoxAlways() As bool
```

Search.ShowSearchOptions Property

Specify "True" to show search options. Set to false to see a very basic UI.

C#

```
[Category("Ektron"), Description("Specify True to show search options. Set to False to see a
very basic UI.")]
public bool ShowSearchOptions;
```

Visual Basic

```
<Category("Ektron"), Description("Specify True to show search options. Set to False to see a
very basic UI.")>
Public Property ShowSearchOptions() As bool
```

Search.Stylesheet Property

Specify the style sheet relative to site root. Leave blank for default.

C#

```
[Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(StringConverter))]
public string Stylesheet;
```

Visual Basic

```
<Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(StringConverter))>
Public Property Stylesheet() As string
```

Search.SyncRoot Property

Gets an object that can be used to synchronize the collection.

C#

```
[Browsable(false)]
public object SyncRoot;
```

Visual Basic

```
<Browsable(false)>
Public Property SyncRoot() As object
```

Search.TargetFolderDescriptionList Property

An optional comma-delimited string, which allows the user to override the folders to search. For example, set to "Photos=1,NewsHeadlines=0" to search only the Photos folder by default, and allow the user to also select the "NewsHeadlines" folder. Also searches these folders' subfolders, returning any matches found there.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("An optional comma-delimited string of folder descriptions, which allows the user
to override which folders to search: For example, set to 'Health Care=1,News=0' to search only
folders with description containing 'Health Care', by default.")]
[Obsolete("This attribute has been deprecated and is no longer supported as of version 6.1")]
[Browsable(false)]
public string TargetFolderDescriptionList;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("An optional comma-delimited string of folder descriptions, which allows the user
to override which folders to search: For example, set to 'Health Care=1,News=0' to search only
folders with description containing 'Health Care', by default.")>
<Obsolete("This attribute has been deprecated and is no longer supported as of version 6.1")>
<Browsable(false)>
Public Property TargetFolderDescriptionList() As string
```

Search.TextBoxSize Property

The width of the search box, in characters.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The width of the search box, in characters.")]
public int TextBoxSize;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The width of the search box, in characters.")>
Public Property TextBoxSize() As Integer
```

Search.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

Sitemap Class

The Sitemap server control utilizes the folder breadcrumb information in the Workarea to display a sitemap of your site. By choosing the starting point of the sitemap, the max levels to display and applying a style class, you can customize the sitemap.

Class Hierarchy

C#

```
[DefaultProperty("Text"), ToolboxData("<{0}:Sitemap runat=\"server\"></{0}:Sitemap>")]
public class Sitemap : EkXsltBase;
```

Visual Basic

```
<DefaultProperty("Text"), ToolboxData("<{0}:Sitemap runat=\"server\"></{0}:Sitemap>")>
Public Class Sitemap
Inherits EkXsltBase
```

File

Sitemap.cs

Sitemap Methods

The methods of the Sitemap class are listed here.

Public Methods

	Name	Description
	Fill (see page 496)	Method that can be overridden and used to grab data from the back end for each server control.

Sitemap.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

Sitemap Properties

The properties of the Sitemap class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 497)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	ClassName (see page 498)	Stylesheet (see page 500) class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.
	DisplayType (see page 498)	Determines how the information on the page is displayed. • **None** - databind only • **List** - lists the title of every content block in the folder • **ListWithDescriptions** - lists the title of every content block in the folder plus the content summary • **Path to Custom Xslt** - enter the path to an Xslt that determines the display of the page
	DisplayXslt (see page 498)	The Xslt to use to render the control (optional).
	FlatTopLevel (see page 499)	Includes site nodes from the parent folder but not their items.
	FolderID (see page 499)	The ID of the folder from which to select data. Root is 0.
	MaxLevel (see page 499)	Get or set the maximum amount of levels to show. Set it to 0 (zero) for unlimited.
	StartingLevel (see page 500)	Get or set to the starting level of the site map. If set to 0 (zero), starts from the root.
	Stylesheet (see page 500)	Specify the location of the style sheet used to display results. Leave blank to use the default style sheet, `\webroot\Workarea\csslib\sitemap.css.`
	XmlDoc (see page 500)	Returns the XML object for the sitemap.

Sitemap.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
```

```
Public virtual Property CacheInterval() As double
```

Sitemap.ClassName Property

Stylesheet (see page 500) class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Browsable(true)]
[Description("Stylesheet class to format the HTML. It must exists in
[site]/workarea/csslib/sitemap.css. Leave blank to accept default.")]
public string ClassName;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Browsable(true)>
<Description("Stylesheet class to format the HTML. It must exists in
[site]/workarea/csslib/sitemap.css. Leave blank to accept default.")>
Public Property ClassName() As string
```

Sitemap.DisplayType Property

Determines how the information on the page is displayed.

- **None** - databind only
- **List** - lists the title of every content block in the folder
- **ListWithDescriptions** - lists the title of every content block in the folder plus the content summary
- **Path to Custom Xslt** - enter the path to an Xslt that determines the display of the page

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Browsable(true)]
[Description("Display type controls how sitemap gets rendered on the page.")]
[Editor(typeof(SitemapDisplayTypeConverter), typeof(System.Drawing.Design.UITypeEditor))]
public string DisplayType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Browsable(true)>
<Description("Display type controls how sitemap gets rendered on the page.")>
<Editor(typeof(SitemapDisplayTypeConverter), typeof(System.Drawing.Design.UITypeEditor))>
Public Property DisplayType() As string
```

Sitemap.DisplayXslt Property

The Xslt to use to render the control (optional).

C#

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("The Xslt to use to render the Control (optional)")]
[TypeConverter(typeof(XsltTypeConverter))]
```

```
[DefaultValue("ecmNavigation")]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("The Xslt to use to render the Control (optional)")>
<TypeConverter(typeof(XsltTypeConverter))>
<DefaultValue("ecmNavigation")>
Public override Property DisplayXslt() As string
```

Sitemap.FlatTopLevel Property

Includes site nodes from the parent folder but not their items.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Includes site nodes from the parent folder but not their items.")]
[DefaultValue(false)]
public bool FlatTopLevel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Includes site nodes from the parent folder but not their items.")>
<DefaultValue(false)>
Public Property FlatTopLevel() As bool
```

Sitemap.FolderID Property

The ID of the folder from which to select data. Root is 0.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The ID of the folder from which to select data. Root is 0.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder from which to select data. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderID() As long
```

Sitemap.MaxLevel Property

Get or set the maximum amount of levels to show. Set it to 0 (zero) for unlimited.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("Set the maximum level of the sitemap. Set it to 0 (zero) for unlimited.")]
public int MaxLevel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("Set the maximum level of the sitemap. Set it to 0 (zero) for unlimited.")>
Public Property MaxLevel() As Integer
```

Sitemap.StartingLevel Property

Get or set to the starting level of the site map. If set to 0 (zero), starts from the root.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Browsable(true)]
[Description("Set the starting level of the sitemap. Set it to 0 for start from root.")]
public int StartingLevel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Browsable(true)>
<Description("Set the starting level of the sitemap. Set it to 0 for start from root.")>
Public Property StartingLevel() As Integer
```

Sitemap.Stylesheet Property

Specify the location of the style sheet used to display results. Leave blank to use the default style sheet, `\webroot\Workarea\csslib\sitemap.css`.

C#

```
[Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(StringConverter))]
public string Stylesheet;
```

Visual Basic

```
<Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(StringConverter))>
Public Property Stylesheet() As string
```

Sitemap.XmlDoc Property

Returns the XML object for the sitemap.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

SmartMenu Class

Class extends the Menu (see page 368)-Sever-Control, adding CSS based menus.

Class Hierarchy

System.Web.UI.WebControls.WebParts.WebPart → Ektron.Cms.Controls.EkWebControl → Ektron.Cms.Controls.EkXsltBase → Ektron.Cms.Controls.Menu → Ektron.Cms.Controls.SmartMenu

C#

```
[DefaultProperty("DefaultMenuID")]
public class SmartMenu : Ektron.Cms.Controls.Menu;
```

Visual Basic

```
<DefaultProperty("DefaultMenuID")>
Public Class SmartMenu
Inherits Ektron.Cms.Controls.Menu
```

File

SmartMenu.cs

SmartMenu Properties

The properties of the SmartMenu class are listed here.

Public Properties

	Name	Description
	AutoCollapseBranches (see page 501)	If you set to **True**, whenever a new submenu opens, all other submenus close. If **False**, other submenus remain open when a new one opens.
	DisplayXslt (see page 502)	The Xslt to use to render the Menu (optional)
	Enable508Compliance (see page 502)	If you set to **True**, the menu will comply with Section 508. This makes the menus accessible to disabled users.
	Enable508NavigationHeadings (see page 502)	Set to **True** to enable accessibility (508-Compliant) navigation headings. This is only used if the Enable508Compliance (see page 502) property is set to True.
	EnableMouseOverPopUp (see page 503)	If you set to **True**, submenus appear as soon as the cursor moves over them. If you set to **False**, submenus only appear if a site visitor clicks them or a keyboard equivalent.
	EnableSmartOpen (see page 503)	If you set to **True**, any submenu on the Smart Menu (see page 368) can be set to open automatically. To learn about how to do this, see the Ektron CMS400.NET Administrator Manual section “Menu (see page 368) Feature” > “Assigning Folders to a Menu (see page 368).” If you set to **False**, even if all other conditions are present, submenus on a Smart Menu (see page 368) do not automatically open. This property lets you design pages whose submenus do not normally open by default. Under some circumstances, such submenus look cluttered.
	EnableSubmenuLinkPathMatching (see page 504)	Set to **True** to enable auto-submenu opening based on link match (if all else fails).
	EnableSubmenuTemplateMatching (see page 504)	Set to **True** to enable auto-submenu opening based on a template filename match (if all else fails).
	LaunchLinksInNewWindow (see page 504)	If you set to **True** and a site visitor selects a menu option, it appears in a new browser window. If **False**, the new page replaces the current page in the browser.
	LogInfo (see page 505)	Set to true to enable logging verbose info to the event log (such as information on reasons for menu selection), must set web.config LogLevel (see page 228) to verbose.
	ShowRootFolder (see page 505)	If you set to **True**, a menu option listing the title of the root folder appears (normally not desired). If you set to **False**, the title of the root folder does not appear.
	SingleEditButton (see page 505)	Set to **True** if you want a single Edit menu item (and no Add menu item) to appear on the menu when you are logged in. If you set to **False**, Edit Menu (see page 368) and Add Menu (see page 368) options appear at the bottom of every menu and submenu.
	StartCollapsed (see page 506)	If you set to **True**, all submenus are closed when the menu initially displays.
	StartWithRootCollapsed (see page 506)	If you set to **True**, all submenus are closed when the menu initially displays.
	Stylesheet (see page 506)	Enter the style sheet that will determine the appearance of the menus. Menu (see page 368) style sheets reside in the `site root/workarea/csslib/smartmenu` folder. You can modify any of the standard style sheets or create your own. (optional)

SmartMenu.AutoCollapseBranches Property

If you set to **True**, whenever a new submenu opens, all other submenus close. If **False**, other submenus remain open when a

new one opens.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to automatically close other menu branches when a new one is
opened.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
public bool AutoCollapseBranches;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to automatically close other menu branches when a new one is
opened.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
Public Property AutoCollapseBranches() As bool
```

SmartMenu.DisplayXslt Property

The Xslt to use to render the Menu (optional)

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override string DisplayXslt;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property DisplayXslt() As string
```

SmartMenu.Enable508Compliance Property

If you set to **True**, the menu will comply with Section 508. This makes the menus accessible to disabled users.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to enable 508 Compliance (make menus accessible to disabled
users).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool Enable508Compliance;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to enable 508 Compliance (make menus accessible to disabled
users).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property Enable508Compliance() As bool
```

SmartMenu.Enable508NavigationHeadings Property

Set to **True** to enable accessibility (508-Compliant) navigation headings. This is only used if the Enable508Compliance (see page 502) property is set to True.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to enable accessibility (508-Compliant) navigation headings. (Only
```

```
used if Enable508Compliance is true.)")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool Enable508NavigationHeadings;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to enable accessibility (508-Compliant) navigation headings. (Only
used if Enable508Compliance is true.)")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property Enable508NavigationHeadings() As bool
```

SmartMenu.EnableMouseOverPopUp Property

If you set to **True**, submenus appear as soon as the cursor moves over them. If you set to **False**, submenus only appear if a site visitor clicks them or a keyboard equivalent.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to enable submenu activation (pop-ups) on mouse-over.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool EnableMouseOverPopUp;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to enable submenu activation (pop-ups) on mouse-over.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property EnableMouseOverPopUp() As bool
```

SmartMenu.EnableSmartOpen Property

If you set to **True**, any submenu on the Smart Menu (see page 368) can be set to open automatically. To learn about how to do this, see the Ektron CMS400.NET Administrator Manual section “Menu (see page 368) Feature” > “Assigning Folders to a Menu (see page 368).”

If you set to **False**, even if all other conditions are present, submenus on a Smart Menu (see page 368) do not automatically open. This property lets you design pages whose submenus do not normally open by default. Under some circumstances, such submenus look cluttered.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("When true, menu will auto-open to show matching IDs or folder associations.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
public bool EnableSmartOpen;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("When true, menu will auto-open to show matching IDs or folder associations.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
Public Property EnableSmartOpen() As bool
```

SmartMenu.EnableSubmenuLinkPathMatching Property

Set to **True** to enable auto-submenu opening based on link match (if all else fails).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to enable auto-submenu opening based on link match (if all else
fails).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool EnableSubmenuLinkPathMatching;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to enable auto-submenu opening based on link match (if all else
fails).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property EnableSubmenuLinkPathMatching() As bool
```

SmartMenu.EnableSubmenuTemplateMatching Property

Set to **True** to enable auto-submenu opening based on a template filename match (if all else fails).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to enable auto-submenu opening based on a template filename match
(if all else fails).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool EnableSubmenuTemplateMatching;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to enable auto-submenu opening based on a template filename match
(if all else fails).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property EnableSubmenuTemplateMatching() As bool
```

SmartMenu.LaunchLinksInNewWindow Property

If you set to **True** and a site visitor selects a menu option, it appears in a new browser window. If **False**, the new page replaces the current page in the browser.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to launch external links in a new window.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool LaunchLinksInNewWindow;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to launch external links in a new window.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property LaunchLinksInNewWindow() As bool
```

SmartMenu.LogInfo Property

Set to true to enable logging verbose info to the event log (such as information on reasons for menu selection), must set web.config LogLevel (see page 228) to verbose.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Used to assist identifying a configuration problem, when the menu does not
appear to select what the designer intended.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool LogInfo;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Used to assist identifying a configuration problem, when the menu does not
appear to select what the designer intended.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property LogInfo() As bool
```

SmartMenu.ShowRootFolder Property

If you set to **True**, a menu option listing the title of the root folder appears (normally not desired). If you set to **False**, the title of the root folder does not appear.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to make the root menu folder-button visible (normally not
desired).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool ShowRootFolder;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to make the root menu folder-button visible (normally not
desired).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property ShowRootFolder() As bool
```

SmartMenu.SingleEditButton Property

Set to **True** if you want a single Edit menu item (and no Add menu item) to appear on the menu when you are logged in. If you set to **False**, Edit Menu (see page 368) and Add Menu (see page 368) options appear at the bottom of every menu and submenu.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to only render a single Edit button (and no Add button) when logged
in.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool SingleEditButton;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to only render a single Edit button (and no Add button) when logged
in.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property SingleEditButton() As bool
```

SmartMenu.StartCollapsed Property

If you set to **True**, all submenus are closed when the menu initially displays.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to initially render menu with submenus closed.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
public bool StartCollapsed;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to initially render menu with submenus closed.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
Public Property StartCollapsed() As bool
```

SmartMenu.StartWithRootCollapsed Property

If you set to **True**, all submenus are closed when the menu initially displays.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to initially render menu with root-menu closed: it's contents made
invisible.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool StartWithRootCollapsed;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to initially render menu with root-menu closed: it's contents made
invisible.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property StartWithRootCollapsed() As bool
```

SmartMenu.Stylesheet Property

Enter the style sheet that will determine the appearance of the menus. Menu (see page 368) style sheets reside in the `site`

`root/workarea/csslib/smartmenu` folder. You can modify any of the standard style sheets or create your own. (optional)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Style-Sheet to use to render the Menu.")]
[TypeConverter(typeof(StyleSheetMenuTypeConverter))]
[DefaultValue("shuttermenu_workarea_vista.css")]
public override string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Style-Sheet to use to render the Menu.")>
<TypeConverter(typeof(StyleSheetMenuTypeConverter))>
<DefaultValue("shuttermenu_workarea_vista.css")>
Public override Property Stylesheet() As string
```

SocialBar Class

Summary description for ListSummary (see page 335).

Class Hierarchy

C#

```
[DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:SocialBar
runat=server></{0}:SocialBar>")]
public class SocialBar : EkXsltWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:SocialBar
runat=server></{0}:SocialBar>")>
Public Class SocialBar
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

File

SocialBar.cs

SocialBar Methods

The methods of the SocialBar class are listed here.

Public Methods

	Name	Description
	Fill (see page 507)	Populates the data for the ListSummary (see page 335).
	GetCallbackResult (see page 508)	Wraps the resulting callback string.
	ToString (see page 508)	Overrides the system ToString function.
	Wrap (see page 508)	Wrap the Html toolbar.

SocialBar.Fill Method

Populates the data for the ListSummary (see page 335).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

SocialBar.GetCallbackResult Method

Wraps the resulting callback string.

C#

```
public string GetCallbackResult();
```

Visual Basic

```
Public Function GetCallbackResult() As string
```

Returns

The callback result string

SocialBar.ToString Method

Overrides the system ToString function.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Returns

String

SocialBar.Wrap Method

Wrap the Html toolbar.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	

SocialBar Properties

The properties of the SocialBar class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 509)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DefaultObjectID (see page 509)	The object ID for the object that the social bar will be drawn for.
	DisplayXslt (see page 510)	The Xslt to use to render the Control (optional).
	DynamicObjectParameter (see page 510)	Checks the QueryString for this value and replaces the list summary with that content block when specified. Leave blank to always display the list summary.
	EkItems (see page 510)	A collection of the buttons on the Social Bar.

	IncludeIcons (see page 511)	Determines whether or not to display an icon next to your navigation lists.
	InviteURL (see page 511)	The URL to use for the Invite (see page 325) item. {2} is the object id and {3} is the object type. If for a group and left blank, will show the group invite window.
	Items (see page 511)	Comma separated list of the items for the social bar.
	MarkupLanguage (see page 512)	Enter name of a markup file that manages the display of this server control.
	MessagingURL (see page 512)	The URL where the messaging control is.
	ObjectType (see page 512)	Sort order key.
	Stylesheet (see page 512)	Specifies the Css to which this control will provide a link.
	SuppressWrapperTags (see page 513)	Suppresses the output of span/div tags around the control.
	XmlDoc (see page 513)	Returns the Xml object loaded for ListSummary (see page 335).

SocialBar.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

SocialBar.DefaultObjectID Property

The object ID for the object that the social bar will be drawn for.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID for the object that the social bar will be drawn for.")]
public long DefaultObjectID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID for the object that the social bar will be drawn for.")>
Public Property DefaultObjectID() As long
```

SocialBar.DisplayXslt Property

The Xslt to use to render the Control (optional).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltSocialBarTypeConverter))]
[DefaultValue("(Path To Custom Xslt)")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltSocialBarTypeConverter))>
<DefaultValue("(Path To Custom Xslt)")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

SocialBar.DynamicObjectParameter Property

Checks the QueryString for this value and replaces the list summary with that content block when specified. Leave blank to always display the list summary.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and uses that value when specified. Leave
blank to always use the default object ID.")]
[Personalizable()]
public string DynamicObjectParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and uses that value when specified. Leave
blank to always use the default object ID.")>
<Personalizable()>
Public Property DynamicObjectParameter() As string
```

SocialBar.EkItems Property

A collection of the buttons on the Social Bar.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("A collection of the buttons on the Social Bar.")]
[Browsable(false)]
public Ektron.Cms.SocialBarItem EkItems;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("A collection of the buttons on the Social Bar.")>
```

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.SocialBarItem
```

SocialBar.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists.

C#

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.")]
[DefaultValue(true)]
[Category("Ektron")]
[Personalizable()]
public bool IncludeIcons;
```

Visual Basic

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.")>
<DefaultValue(true)>
<Category("Ektron")>
<Personalizable()>
Public Property IncludeIcons() As bool
```

SocialBar.InviteURL Property

The URL to use for the Invite (see page 325) item. {2} is the object id and {3} is the object type. If for a group and left blank, will show the group invite window.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("The URL to use for the Invite item. {2} is the object id and {3} is the object
type. If for a group and left blank, will show the group invite window.")]
public string InviteURL;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("The URL to use for the Invite item. {2} is the object id and {3} is the object
type. If for a group and left blank, will show the group invite window.")>
Public Property InviteURL() As string
```

SocialBar.Items Property

Comma separated list of the items for the social bar.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Addto, Email, Digg, Delicious, Print, Twitter")]
[Description("Comma separated list of the items for the social bar.")]
public string Items;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Addto, Email, Digg, Delicious, Print, Twitter")>
<Description("Comma separated list of the items for the social bar.")>
Public Property Items() As string
```

SocialBar.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

SocialBar.MessagingURL Property

The URL where the messaging control is.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("The URL where the messaging control is.")]
public string MessagingURL;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("The URL where the messaging control is.")>
Public Property MessagingURL() As string
```

SocialBar.ObjectType Property

Sort order key.

C#

```
[Bindable(true)]
[DefaultValue(Ektron.Cms.Common.EkEnumeration.CMSSocialBarTypes.Content)]
[Description("The type of the object that the social bar will be drawn for.")]
[Category("Ektron")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Order"), WebDescription("Order")]
public Ektron.Cms.Common.EkEnumeration.CMSSocialBarTypes ObjectType;
```

Visual Basic

```
<Bindable(true)>
<DefaultValue(Ektron.Cms.Common.EkEnumeration.CMSSocialBarTypes.Content)>
<Description("The type of the object that the social bar will be drawn for.")>
<Category("Ektron")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Order"), WebDescription("Order")>
Public Property ObjectType() As Ektron.Cms.Common.EkEnumeration.CMSSocialBarTypes
```

SocialBar.Stylesheet Property

Specifies the Css to which this control will provide a link.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
```

```
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")>
<Personalizable()>
Public Property Stylesheet() As string
```

SocialBar.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

C#

```
public override bool SuppressWrapperTags;
```

Visual Basic

```
Public override Property SuppressWrapperTags() As bool
```

SocialBar.XmlDoc Property

Returns the Xml object loaded for ListSummary (see page 335).

C#

```
public override XmlDocument XmlDoc;
```

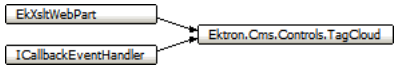
Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

TagCloud Class

The TagCloud server control allows you to visually depict a weighted list of tags assigned to users or community groups. The weight of each tag is determined by the amount of times a tag is assigned and represented by font size. The large the font size the more times a tag is assigned. Tags (see page 805) are keywords that are associated with a user or community group in CMS400.NET.

Class Hierarchy



C#

```
[DefaultProperty(""), ToolboxData("<{0}:TagCloud runat=server></{0}:TagCloud>")]
public class TagCloud : EkXsltWebPart, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty(""), ToolboxData("<{0}:TagCloud runat=server></{0}:TagCloud>")>
Public Class TagCloud
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

File

TagCloud.cs

TagCloud Methods

The methods of the TagCloud class are listed here.

Public Methods

	Name	Description
	Fill (see page 514)	Populates the data for the Favorites (see page 236).
	ToString (see page 514)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 514)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

TagCloud.Fill Method

Populates the data for the Favorites (see page 236).

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

TagCloud.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

TagCloud.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

TagCloud Properties

The properties of the TagCloud class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 515)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

	DisplayXslt (see page 516)	Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.(optional).
	DynamicParameter (see page 516)	
	MarkupLanguage (see page 516)	Identify the template markup file that controls the display of the server control.
	MaxTagsDisplayed (see page 517)	The maximum number of tags to display. Set to 0 (zero) for unlimited. The tags that are displayed are always the top tags based on usage. For example, if the number is ten, the ten most used tags are displayed. When using the OrderBy (see page 517) property in conjunction with this property, it gets the given number of tags and then sorts them alphabetically.
	OrderBy (see page 517)	The criteria by which the tags are ordered. Choices are: • **TaggedCount** - sort by how many times the tags has been assigned. • **Text** - sort the list in order A to Z.
	OrderByDirection (see page 517)	Select the direction of the TagOrderBy property. Choose Ascending or Descending.
	Stylesheet (see page 518)	Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet.
	TagTemplate (see page 518)	Add a path to another Web form to create links for the tag text. This path can be relative or absolute. There are five parameters that are automatically added to the link's QueryString that allow you to pass information about tag. • **searchtag** or **searchgrptag** - searchtag represents the tag's text for users and searchgrptag represent the tag's text for community groups. When the template contains a CommunitySearch (see page 135) server control, the control sees these parameters and performs a search based on the text. • **TagId** - represents the tag's ID. • **TagLanguage** - represent the tags language. • **TagCount** - represent the tag's count.... more (see page 518)
	TagTemplateTarget (see page 519)	Determines the type of window that appears on this Web form when a user clicks an item in the TagCloud (see page 513) control. The default is _self.
	TagType (see page 519)	Select whether the control displays tags for a user or a community group. Choices are: • **User (see page 950)** - show tags that are assigned to users. • **CommunityGroup (see page 750)** - show tags that are assigned to community group. • **Content (see page 822)** - show tags that are assigned to content. • **Library (see page 695)** - show tags that are assigned to library items.
	XmlDoc (see page 519)	Returns the Xslt object for the collection.

TagCloud.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

TagCloud.DisplayXslt Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.(optional).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute. (optional)")]
[TypeConverter(typeof(XsltFavoriteTypeConverter))]
[DefaultValue("Builtin")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute. (optional)")>
<TypeConverter(typeof(XsltFavoriteTypeConverter))>
<DefaultValue("Builtin")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

TagCloud.DynamicParameter Property

C#

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("")]
[DefaultValue("id")]
public string DynamicParameter;
```

Visual Basic

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("")>
<DefaultValue("id")>
Public Property DynamicParameter() As string
```

TagCloud.MarkupLanguage Property

Identify the template markup file that controls the display of the server control.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Identify the template
markup file that controls the display of the server control.")]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Identify the template
```

```
markup file that controls the display of the server control.")>
Public Property MarkupLanguage() As string
```

TagCloud.MaxTagsDisplayed Property

The maximum number of tags to display. Set to 0 (zero) for unlimited. The tags that are displayed are always the top tags based on usage. For example, if the number is ten, the ten most used tags are displayed. When using the OrderBy (see page 517) property in conjunction with this property, it gets the given number of tags and then sorts them alphabetically.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The maximum number of tags to display. Set to 0 (zero) for unlimited. The tags
that are displayed are always the top tags based on usage. For example, if the number is ten,
the ten most used tags are displayed. When using the OrderBy property in conjunction with this
property, it gets the given number of tags and then sorts them alphabetically.")]
[DefaultValue("50")]
public int MaxTagsDisplayed;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The maximum number of tags to display. Set to 0 (zero) for unlimited. The tags
that are displayed are always the top tags based on usage. For example, if the number is ten,
the ten most used tags are displayed. When using the OrderBy property in conjunction with this
property, it gets the given number of tags and then sorts them alphabetically.")>
<DefaultValue("50")>
Public Property MaxTagsDisplayed() As Integer
```

TagCloud.OrderBy Property

The criteria by which the tags are ordered. Choices are:

- **TaggedCount** - sort by how many times the tags has been assigned.
- **Text** - sort the list in order A to Z.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Text")]
[TypeConverter(typeof(TagOrderByTypeConverter))]
[Description("The criteria by which the tags are ordered. Choices are TaggedCount and Text")]
public string OrderBy;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Text")>
<TypeConverter(typeof(TagOrderByTypeConverter))>
<Description("The criteria by which the tags are ordered. Choices are TaggedCount and Text")>
Public Property OrderBy() As string
```

TagCloud.OrderByDirection Property

Select the direction of the TagOrderBy property. Choose Ascending or Descending.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Select the direction of the TagOrderBy property. Choose Ascending or
Descending.")]
[TypeConverter(typeof(OrderByDirectionTypeConverter))]
[DefaultValue("Ascending")]
```

```
public string OrderByDirection;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Select the direction of the TagOrderBy property. Choose Ascending or
Descending.")>
<TypeConverter(typeof(OrderByDirectionTypeConverter))>
<DefaultValue("Ascending")>
Public Property OrderByDirection() As string
```

TagCloud.Stylesheet Property

Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Specify the path to a style sheet for use with the sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Specify the path to a style sheet for use with the sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<Personalizable()>
Public Property Stylesheet() As string
```

TagCloud.TagTemplate Property

Add a path to another Web form to create links for the tag text. This path can be relative or absolute. There are five parameters that are automatically added to the link's QueryString that allow you to pass information about tag.

- **searchtag** or **searchgrptag** - searchtag represents the tag's text for users and searchgrptag represent the tag's text for community groups. When the template contains a CommunitySearch (see page 135) server control, the control sees these parameters and performs a search based on the text.
- **TagId** - represents the tag's ID.
- **TagLanguage** - represent the tags language.
- **TagCount** - represent the tag's count. The tag's count is the amount of times a tag has been used.
- **TagType (see page 519)** - represents the tag's type. The tag's type will be user or community group.

In addition to these parameters, you can add your own by defining them in the path. When you do, these parameters will be appended to your parameters.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(""), Description("Add a path to another Web
form to create links for the tag text. This path can be relative or absolute. There are five
parameters that are automatically added to the link?s QueryString that allow you to pass
information about tag. They are searchtag or searchgrptag depending whether the control is set
to user or group and TagId, TagLanguage, TagCount and TagType.") ]
public string TagTemplate;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(""), Description("Add a path to another Web
form to create links for the tag text. This path can be relative or absolute. There are five
parameters that are automatically added to the link?s QueryString that allow you to pass
information about tag. They are searchtag or searchgrptag depending whether the control is set
to user or group and TagId, TagLanguage, TagCount and TagType.") >
Public Property TagTemplate() As string
```

TagCloud.TagTemplateTarget Property

Determines the type of window that appears on this Web form when a user clicks an item in the TagCloud (see page 513) control. The default is _self.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self),
Description("Determines the type of window that appears on this Web form when a user clicks an
item in the TagCloud control. The default is _self.") ]
public ItemLinkTargets TagTemplateTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self),
Description("Determines the type of window that appears on this Web form when a user clicks an
item in the TagCloud control. The default is _self.") >
Public Property TagTemplateTarget() As ItemLinkTargets
```

TagCloud.TagType Property

Select whether the control displays tags for a user or a community group. Choices are:

- **User (see page 950)** - show tags that are assigned to users.
- **CommunityGroup (see page 750)** - show tags that are assigned to community group.
- **Content (see page 822)** - show tags that are assigned to content.
- **Library (see page 695)** - show tags that are assigned to library items.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Select whether the control displays tags for a user, community group, content,
or library item. Choices are User, CommunityGroup, Content, and Library.")]
[TypeConverter(typeof(TagTypeTypeConverter))]
[DefaultValue("User")]
public string TagType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Select whether the control displays tags for a user, community group, content,
or library item. Choices are User, CommunityGroup, Content, and Library.")>
<TypeConverter(typeof(TagTypeTypeConverter))>
<DefaultValue("User")>
Public Property TagType() As string
```

TagCloud.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

UserProfile Class

Class Hierarchy

C#

```
[DefaultProperty("DefaultUserID"), ToolboxData("<{0}:UserProfile
runat=server></{0}:UserProfile>")]
public class UserProfile : EkXsltWebPart;
```

Visual Basic

```
<DefaultProperty("DefaultUserID"), ToolboxData("<{0}:UserProfile
runat=server></{0}:UserProfile>")>
Public Class UserProfile
Inherits EkXsltWebPart
```

File

Profile.cs

UserProfile Methods

The methods of the UserProfile class are listed here.

Public Methods

	Name	Description
	Fill (see page 520)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 520)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 520)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

UserProfile.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

UserProfile.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

UserProfile.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar.

Returns

UserProfile Properties

The properties of the UserProfile class are listed here.

Public Properties

	Name	Description
	AvatarHeight (see page 521)	Gets or Sets the avatar height. Use 0 for the actual image dimensions.
	AvatarWidth (see page 521)	Gets or Sets the avatar width. Use 0 for the actual image dimensions.
	CacheInterval (see page 522)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	DefaultUserID (see page 522)	Gets or Sets the default user ID for this control. This is the user ID that will be shown if the DynamicParameter (see page 523) does not get a value. (e.g. userprofile.aspx?userid=2 will get the value 2 for a DynamicParameter (see page 523) of "id", and will display DefaultUserID otherwise.)
	DisplayXslt (see page 523)	The Xslt to use to render the control (optional).
	DynamicParameter (see page 523)	To make this control dynamic, select ID. When you do, this server control is attached to the user passed as a URL parameter.
	EnableEmailDisplay (see page 523)	Enables or disables showing the users email in the profile information.
	Stylesheet (see page 524)	Specifies the Css to which this control will provide a link.Select either Use Default, None, or enter a path to a custom Css file.
	TagTemplate (see page 524)	Gets or Sets the template-url for the personal tag links.
	TagTemplateTarget (see page 524)	Gets or Sets the target (window) attribute for the personal-tag links.
	TaxonomyId (see page 524)	Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.
	XmlDoc (see page 525)	Returns the Xslt object for the collection.

UserProfile.AvatarHeight Property

Gets or Sets the avatar height. Use 0 for the actual image dimensions.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(74), Description("Gets or Sets the avatar
height. Use 0 for the actual image dimensions.") ]
public int AvatarHeight;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(74), Description("Gets or Sets the avatar
height. Use 0 for the actual image dimensions.") >
Public Property AvatarHeight() As Integer
```

UserProfile.AvatarWidth Property

Gets or Sets the avatar width. Use 0 for the actual image dimensions.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(74), Description("Gets or Sets the avatar
width. Use 0 for the actual image dimensions.") ]
public int AvatarWidth;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(74), Description("Gets or Sets the avatar
width. Use 0 for the actual image dimensions.") >
Public Property AvatarWidth() As Integer
```

UserProfile.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

UserProfile.DefaultUserID Property

Gets or Sets the default user ID for this control. This is the user ID that will be shown if the DynamicParameter (see page 523) does not get a value. (e.g. userprofile.aspx?userid=2 will get the value 2 for a DynamicParameter (see page 523) of "id", and will display DefaultUserID otherwise.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Gets or sets the default user ID for this control. This is the user ID that will
be shown if the DynamicParameter does not get a value. (e.g. userprofile.aspx?userid=2 will
get the value 2 for a DynamicParameter of \"id\", and will use DefaultUserID otherwise.)")]
public long DefaultUserID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Gets or sets the default user ID for this control. This is the user ID that will
be shown if the DynamicParameter does not get a value. (e.g. userprofile.aspx?userid=2 will
get the value 2 for a DynamicParameter of \"id\", and will use DefaultUserID otherwise.)")>
Public Property DefaultUserID() As long
```

UserProfile.DisplayXslt Property

The Xslt to use to render the control (optional).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltProfileTypeConverter))]
[DefaultValue("ProfileBasic.xsl")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltProfileTypeConverter))>
<DefaultValue("ProfileBasic.xsl")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

UserProfile.DynamicParameter Property

To make this control dynamic, select ID. When you do, this server control is attached to the user passed as a URL parameter.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("To make this control dynamic, select ID. When you do, this server control is
attached to the user passed as a URL parameter.")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("To make this control dynamic, select ID. When you do, this server control is
attached to the user passed as a URL parameter.")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

UserProfile.EnableEmailDisplay Property

Enables or disables showing the users email in the profile information.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("false")]
[TypeConverter(typeof(bool))]
[Description("Set to true to display the users email address in the profile info, set to false
to hide this (default)")]
[Personalizable()]
public bool EnableEmailDisplay;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("false")>
<TypeConverter(typeof(bool))>
```

```
<Description("Set to true to display the users email address in the profile info, set to false
to hide this (default)")>
<Personalizable()>
Public Property EnableEmailDisplay() As bool
```

UserProfile.Stylesheet Property

Specifies the Css to which this control will provide a link.Select either Use Default, None, or enter a path to a custom Css file.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")>
<Personalizable()>
Public Property Stylesheet() As string
```

UserProfile.TagTemplate Property

Gets or Sets the template-url for the personal tag links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a user
search, when a tag is clicked (example: UserSearch.aspx).") ]
public string TagTemplate;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a user
search, when a tag is clicked (example: UserSearch.aspx).") >
Public Property TagTemplate() As string
```

UserProfile.TagTemplateTarget Property

Gets or Sets the target (window) attribute for the personal-tag links.

C#

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the personal tag links (_blank, _self, _parent or _top), to select opening
in the same window, a popup, etc.") ]
public ItemLinkTargets TagTemplateTarget;
```

Visual Basic

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the personal tag links (_blank, _self, _parent or _top), to select opening
in the same window, a popup, etc.") >
Public Property TagTemplateTarget() As ItemLinkTargets
```

UserProfile.TaxonomyId Property

Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)]
[Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")]
[Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long TaxonomyId;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)>
<Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")>
<Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property TaxonomyId() As long
```

UserProfile.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

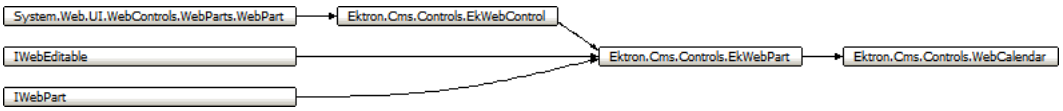
Visual Basic

```
Public override Property XmlDoc() As XmlDocument
```

WebCalendar Class

Summary description for Calendar (see page 76).

Class Hierarchy



C#

```
[ToolboxData("<{0}:WebCalendar runat=server></{0}:WebCalendar>")]
[ParseChildren(true)]
public class WebCalendar : EkWebPart;
```

Visual Basic

```
<ToolboxData("<{0}:WebCalendar runat=server></{0}:WebCalendar>")>
<ParseChildren(true)>
Public Class WebCalendar
Inherits EkWebPart
```

File

WebCalendar.cs

WebCalendar Methods

The methods of the WebCalendar class are listed here.

Public Methods

	Name	Description
	Fill (see page 526)	Method that can be overridden and used to grab data from the back end for each server control.
	Wrap (see page 526)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

WebCalendar.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

WebCalendar.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

WebCalendar Properties

The properties of the WebCalendar class are listed here.

Public Properties

	Name	Description
	AllowEventEditing (see page 526)	Allow Editing
	CssFile (see page 527)	Override the default Css file to include (Workarea/WebCalendar (see page 525)/View/WebCalendar.css) to render the calendar. This does not affect the Event form. this should be an absolute or relative url.
	DataSource (see page 527)	The datasources for the calendar
	DefaultDisplayType (see page 527)	Specify the default Calendar (see page 76) view.
	DisplayTemplatePath (see page 527)	Virtual Path to User (see page 950) Control that will be used to display events on the calendar. defaults to ApplicationPath + "WebCalendarDefaultTemplateDisplay.ascx"
	DisplayType (see page 528)	Specify the allowed Calendar (see page 76) view.
	DynamicParameter (see page 528)	The QueryString to check for a dynamic ID.
	RequestInformationRef (see page 528)	Retrieve the current Request Information object for this request.
	Text (see page 529)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

WebCalendar.AllowEventEditing Property

Allow Editing

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Specify whether or not to allow editing of events.")]
```

```
public bool AllowEventEditing;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Specify whether or not to allow editing of events.")>
Public Property AllowEventEditing() As bool
```

WebCalendar.CssFile Property

Override the default Css file to include (Workarea/WebCalendar (see page 525)/View/WebCalendar.css) to render the calendar. This does not affect the Event form. this should be an absolute or relative url.

C#

```
public string CssFile;
```

Visual Basic

```
Public Property CssFile() As string
```

WebCalendar.DataSource Property

The datasources for the calendar

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The datasources to be displayed.")]
[DesignerSerializationVisibility(DesignerSerializationVisibility.Content)]
[PersistenceMode(PersistenceMode.InnerProperty)]
public List<CalendarDataSource> DataSource;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The datasources to be displayed.")>
<DesignerSerializationVisibility(DesignerSerializationVisibility.Content)>
<PersistenceMode(PersistenceMode.InnerProperty)>
Public Property DataSource() As List<CalendarDataSource>
```

WebCalendar.DefaultDisplayType Property

Specify the default Calendar (see page 76) view.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(DefaultDisplayStyle.Month)]
[Description("Specify the default Calendar view.")]
[Personalizable()]
public DefaultDisplayStyle DefaultDisplayType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(DefaultDisplayStyle.Month)>
<Description("Specify the default Calendar view.")>
<Personalizable()>
Public Property DefaultDisplayType() As DefaultDisplayStyle
```

WebCalendar.DisplayTemplatePath Property

Virtual Path to User (see page 950) Control that will be used to display events on the calendar. defaults to ApplicationPath +

"WebCalendarDefaultTemplateDisplay.ascx"

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Workarea\\WebCalendar\\DefaultTemplate\\Display.ascx")]
[Description("Virtual Path to User Control that will be used to display events on the
calendar. defaults to ApplicationPath + \"\\WebCalendar\\DefaultTemplate\\Display.ascx\"")]
public string DisplayTemplatePath;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Workarea\\WebCalendar\\DefaultTemplate\\Display.ascx")>
<Description("Virtual Path to User Control that will be used to display events on the
calendar. defaults to ApplicationPath + \"\\WebCalendar\\DefaultTemplate\\Display.ascx\"")>
Public Property DisplayTemplatePath() As string
```

WebCalendar.DisplayType Property

Specify the allowed Calendar (see page 76) view.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(DisplayStyle.All)]
[Description("Specify the allowed Calendar view.")]
[Personalizable()]
public DisplayStyle DisplayType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(DisplayStyle.All)>
<Description("Specify the allowed Calendar view.")>
<Personalizable()>
Public Property DisplayType() As DisplayStyle
```

WebCalendar.DynamicParameter Property

The QueryString to check for a dynamic ID.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The QueryString to check for a dynamic ID.")]
[TypeConverter(typeof(DynamicCalendarTypeConverter))]
[DefaultValue("None - Use Default")]
[Personalizable()]
public string DynamicParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The QueryString to check for a dynamic ID.")>
<TypeConverter(typeof(DynamicCalendarTypeConverter))>
<DefaultValue("None - Use Default")>
<Personalizable()>
Public Property DynamicParameter() As string
```

WebCalendar.RequestInformationRef Property

Retrieve the current Request Information object for this request.

C#

```
public EkRequestInformation RequestInformationRef;
```

Visual Basic

```
Public Property RequestInformationRef() As EkRequestInformation
```

WebCalendar.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

C#

```
public override string Text;
```

Visual Basic

```
Public override Property Text() As string
```

WebSearch Class

The Web Search (see page 474) server control lets you customize the behavior of the search. You place this control on any Web form from which a site visitor can search your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 725)" > "Web Site (see page 725) Search (see page 474)."

Class Hierarchy

C#

```
[DefaultProperty("FolderID"), ToolboxData("<{0}:WebSearch runat=server></{0}:WebSearch>")]
public class WebSearch : EkXsltBase, ICallbackEventHandler;
```

Visual Basic

```
<DefaultProperty("FolderID"), ToolboxData("<{0}:WebSearch runat=server></{0}:WebSearch>")>
Public Class WebSearch
Inherits EkXsltBase
Inherits ICallbackEventHandler
```

File

WebSearch.cs

WebSearch.WebSearch Constructor

Constructor for the WebSearch class.

C#

```
public WebSearch();
```

Visual Basic

```
Public Sub New()
```

WebSearch Methods

The methods of the WebSearch class are listed here.

Public Methods

	Name	Description
	Fill (see page 530)	Method that can be overridden and used to grab data from the back end for each server control.
	GetCallbackResult (see page 530)	Returns search results to the client callback function
	LoadSearch (see page 530)	Search (see page 474) HTML, content, images, multimedia and documents based on content or metadata. This method can be used to perform programmatic search.
	RaiseCallbackEvent (see page 530)	Processes the callback event by performing the search.
	Wrap (see page 531)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

WebSearch.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

C#

```
public override void Fill();
```

Visual Basic

```
Public override Function Fill() As void
```

WebSearch.GetCallbackResult Method

Returns search results to the client callback function

C#

```
public string GetCallbackResult();
```

Visual Basic

```
Public Function GetCallbackResult() As string
```

Returns

String containing search results.

WebSearch.LoadSearch Method

Search (see page 474) HTML, content, images, multimedia and documents based on content or metadata. This method can be used to perform programmatic search.

C#

```
public int LoadSearch(SearchRequestData searchReqData);
```

Visual Basic

```
Public Function LoadSearch(searchReqData As SearchRequestData) As Integer
```

Parameters

Parameters	Description
SearchRequestData searchReqData	Search (see page 474) criteria bundled in a SearchRequestData object.

Returns

Number of records satisfying search criteria

WebSearch.RaiseCallbackEvent Method

Processes the callback event by performing the search.

C#

```
public void RaiseCallbackEvent(string eventArgs);
```

Visual Basic

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

Parameters

Parameters	Description
string eventArgs	Name-value pairs representing form fields passed by client callback function

WebSearch.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

C#

```
public override string Wrap(string Html);
```

Visual Basic

```
Public override Function Wrap(Html As string) As string
```

Parameters

Parameters	Description
string Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

WebSearch Properties

The properties of the WebSearch class are listed here.

Public Properties

	Name	Description
	ButtonImgSrc (see page 534)	Optional: If you want to display an image on the submit button, enter the server path to it.
	ButtonText (see page 534)	The text used for the button if no image source is identified. If an image source is identified, this is alternative text for the button.
	CustomOrderBy (see page 534)	Order search results by a custom property. Set this property to the friendly name of the property defined in the Indexing Service. If an invalid property is specified, no search results are returned. If this is left empty, the OrderBy (see page 539) property is used to order search results.
	CustomSearch (see page 535)	Optional: If you want to display an image on the submit button, enter the server path to it.
	DisableForumSearch (see page 535)	Enable Forum (see page 264) Search (see page 474).
	DisplayXslt (see page 535)	Determines the display of the search results page. • **None** - databind only • **ecmNavigation** - lists the title of every content item found by the search • **ecmTeaser** - lists the title and the summary of every content item found by the search • **ecmUnOrderedList** - sorts the list in no particular order. Shows the title and content summary • **Path to Custom Xslt** - enter the path to an Xslt that determines the display of the page
	DynamicContentTemplate (see page 536)	Get or sets the template of dynamic content. If specified control will not use dynamic box.

	EnableAdvancedLink (see page 536)	Set to **True** to display the link that makes the advanced properties visible.
	EndCount (see page 536)	Get the numerical record of the first item on a page.
	FolderID (see page 536)	The folder at which the search begins. The starting folder need not be the root folder. The **Recursive (see page 540)** property determines if the search examines this folder's subfolders.
	IsInWorkArea (see page 537)	Is server control in workarea
	LinkTarget (see page 537)	Gets or sets the target window for the hyperlinks in the search result. The choices are: • **_blank** - loads the link in a new blank window. This window is not named. • **_self** - loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target. • **_parent** - loads the link in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** - loads the link in the full body of the window. This defaults to acting like "_self" if the... more (see page 537)
	MarkupLanguage (see page 538)	Identify the template markup file that controls the display of the search results. For example, `mysearchmarkup.ekml`. Enter the path relative to site root. For example, `\workarea\customfiles\markup\mysearchmarkup.ekml`. If you enter a valid EkML file, the DisplayXslt (see page 535) property value is ignored.
	MaxCharacters (see page 538)	The maximum number of characters the Search (see page 474) text box accepts. If you enter a value less than 50, it is a good idea to set the TextBoxSize (see page 547) property to the same number.
	MaxTeaserLength (see page 538)	An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary is truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT).
	MultipleFolders (see page 539)	Optional: If you want to display an image on the submit button, enter the server path to it.
	OrderBy (see page 539)	The order of the search results. Choices are: • **Title** - The content title (alphabetical) • **ID** - The content ID number • **Date Created** - The date the content was created • **Date Modified** - The date the content was most recently modified • **Editor** - The user who last edited the content (alphabetical) • **Rank** - The rank assigned to the content. See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 725)" > "Display of Search (see page 474) Results" > "Search (see page 474) Result Ranking"
	OrderDirection (see page 539)	Specify the order direction of the results. Choices are **Ascending** or **Descending**.
	Recursive (see page 540)	Select **True** if you wish to search sub-folders of the starting folder. The starting folder is identified in the `FolderID (see page 536)` property.
	RemoveTeaserHtml (see page 540)	An optional result formatting parameter. If set to **True**, any HTML in the content items' teaser/summary is removed before it is sent to the browser. If set to **False**, any existing HTML is passed along, (this may then be further filtered by the XSLT).
	ResultCount (see page 540)	The number of search results. This is a read-only property.
	ResultDataTable (see page 541)	Allows a developer to get results from the search control as a DataTable
	ResultsPageSize (see page 541)	An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary is truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT).
	ResultTagId (see page 542)	Lets you designate where search results appear. You can place search criteria in one area of a Web form and the results in another. For example, you have the following tag. `<span id="results"></span>` In this case, enter **results** for the property value.
	ResultXml (see page 542)	Allows a developer to get results from the search control as XML.

	SearchFor (see page 542)	Choose the type of content to search. • All • HTML • Documents • Images • Multimedia • Discussion Forums • Tags (see page 805)
	SearchSynonyms (see page 543)	If set to **True**, the search results will search for synonyms configured for term. The default is True. • **True** - Display search for synonym of term specified. • **False** - Do not search synonyms.
	SearchText (see page 543)	Get or set default search text. This is optional.
	ShowCategories (see page 544)	If set to **True**, taxonomy categories will also be searched for the search text specified. The default is False. **True** = Search (see page 474) across taxonomy categories **False** = Do not search across taxonomy categories
	ShowCustomSummary (see page 544)	If set to **True**, custom summary is used in search results. The default is False. **True** = Display custom summary in search results. **False** = Display characterization generated by indexing service in search results.
	ShowSearchBoxAlways (see page 544)	If set to **False**, the search box does not appear on PostBack. The default is True. **True** = Display search box on PostBack **False** = Do not display search box on PostBack
	ShowSearchOptions (see page 545)	If set to **True**, the following options appear above the Search (see page 474) box. • Site (see page 725) • Html • Documents • Images • Multimedia • Tags (see page 805)
	ShowSuggestedResults (see page 545)	If set to **True**, the search results include Suggested Results if any exist. The default is True. • **True** = Display Suggested Results • **False** = Do not display Suggested Results
	StartCount (see page 546)	Get the numerical record of the last item on a page.
	Stylesheet (see page 546)	Specify the location of the style sheet for the search results page. Set the location relative to the site root folder. For example: `csslib\mytest.css`. Leave blank to use the default style sheet,
	TaxCategories (see page 546)	Is server control in workarea
	TaxonomyOperator (see page 546)	Taxonomy (see page 876) operator used in filtering search results by taxonomy category. • and - selected taxonomy categories are AND-ed • or - selected taxonomy categories are OR-ed
	TextBoxSize (see page 547)	The size of the Search (see page 474) text box for user input. The width is based on the number of characters.
	TimeTaken (see page 547)	Gets the amount of time, in seconds, it has taken to perform the search.
	XmlDoc (see page 547)	This property is not supported by WebSearch (see page 529). The underlying XML can be displayed using an XSLT specified by the DisplayXslt (see page 535) property of the WebSearch (see page 529) server control. The XSLT filename is specified relative to the current directory.

WebSearch.ButtonImgSrc Property

Optional: If you want to display an image on the submit button, enter the server path to it.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Optional: The path to the image you wish to use for the button. This path is
assumed to be relative to workarea/images.")]
public string ButtonImgSrc;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Optional: The path to the image you wish to use for the button. This path is
assumed to be relative to workarea/images.")>
Public Property ButtonImgSrc() As string
```

WebSearch.ButtonText Property

The text used for the button if no image source is identified. If an image source is identified, this is alternative text for the button.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[Description("The text of the search button if no image is specified. The default value is set
to \"res_srch_btn\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_SEARCH_BTN)]
public string ButtonText;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<Description("The text of the search button if no image is specified. The default value is set
to \"res_srch_btn\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_SEARCH_BTN)>
Public Property ButtonText() As string
```

WebSearch.CustomOrderBy Property

Order search results by a custom property. Set this property to the friendly name of the property defined in the Indexing Service. If an invalid property is specified, no search results are returned. If this is left empty, the OrderBy (see page 539) property is used to order search results.

C#

```
[Category("Ektron"), DefaultValue(""), Description("Order search results by a custom property.
Set this property to the friendly name of the property.")]
public string CustomOrderBy;
```

Visual Basic

```
<Category("Ektron"), DefaultValue(""), Description("Order search results by a custom property.
Set this property to the friendly name of the property.")>
Public Property CustomOrderBy() As string
```

WebSearch.CustomSearch Property

Optional: If you want to display an image on the submit button, enter the server path to it.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Optional: Custom directories to search.")]
public string CustomSearch;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Optional: Custom directories to search.")>
Public Property CustomSearch() As string
```

WebSearch.DisableForumSearch Property

Enable Forum (see page 264) Search (see page 474).

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true Discussion Forums option does not appear in search.")]
[Personalizable(), WebBrowsable()]
public bool DisableForumSearch;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true Discussion Forums option does not appear in search.")>
<Personalizable(), WebBrowsable()>
Public Property DisableForumSearch() As bool
```

WebSearch.DisplayXslt Property

Determines the display of the search results page.

- **None** - databind only
- **ecmNavigation** - lists the title of every content item found by the search
- **ecmTeaser** - lists the title and the summary of every content item found by the search
- **ecmUnOrderedList** - sorts the list in no particular order. Shows the title and content summary
- **Path to Custom Xslt** - enter the path to an Xslt that determines the display of the page

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the Control (optional)")]
[TypeConverter(typeof(XsltTypeConverter))]
[DefaultValue("None - Databind Only")]
[Personalizable(), WebBrowsable()]
```

```
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the Control (optional)")>
<TypeConverter(typeof(XsltTypeConverter))>
<DefaultValue("None - Databind Only")>
<Personalizable(), WebBrowsable()>
Public override Property DisplayXslt() As string
```

WebSearch.DynamicContentTemplate Property

Get or sets the template of dynamic content. If specified control will not use dynamic box.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Get or sets the template of dynamic content. Overwrites the content quicklink.")]
public string DynamicContentTemplate;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Get or sets the template of dynamic content. Overwrites the content quicklink.")>
Public Property DynamicContentTemplate() As string
```

WebSearch.EnableAdvancedLink Property

Set to **True** to display the link that makes the advanced properties visible.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the link that makes the advanced properties visible.")]
public bool EnableAdvancedLink;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the link that makes the advanced properties visible.")>
Public Property EnableAdvancedLink() As bool
```

WebSearch.EndCount Property

Get the numerical record of the first item on a page.

C#

```
[Browsable(false)]
public int EndCount;
```

Visual Basic

```
<Browsable(false)>
Public Property EndCount() As Integer
```

WebSearch.FolderID Property

The folder at which the search begins. The starting folder need not be the root folder. The **Recursive (see page 540)** property

determines if the search examines this folder's subfolders.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("The ID of the folder you wish to select data from. Root is 0.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long FolderID;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The ID of the folder you wish to select data from. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property FolderID() As long
```

WebSearch.IsInWorkArea Property

Is server control in workarea

C#

```
[Browsable(false)]
public bool IsInWorkArea;
```

Visual Basic

```
<Browsable(false)>
Public Property IsInWorkArea() As bool
```

WebSearch.LinkTarget Property

Gets or sets the target window for the hyperlinks in the search result. The choices are:

- **_blank** - loads the link in a new blank window. This window is not named.
- **_self** - loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.
- **_parent** - loads the link in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.
- **_top** - loads the link in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(ItemLinkTargets._self)]
[Description("The target attribute of the link.")]
public ItemLinkTargets LinkTarget;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(ItemLinkTargets._self)>
<Description("The target attribute of the link.")>
Public Property LinkTarget() As ItemLinkTargets
```

WebSearch.MarkupLanguage Property

Identify the template markup file that controls the display of the search results. For example, `mysearchmarkup.ekml`.

Enter the path relative to site root. For example, `\workarea\customfiles\markup\mysearchmarkup.ekml`.

If you enter a valid EkML file, the DisplayXslt (see page 535) property value is ignored.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Enter name of a markup file that manages the display of this server control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Enter name of a markup file that manages the display of this server control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

WebSearch.MaxCharacters Property

The maximum number of characters the Search (see page 474) text box accepts. If you enter a value less than 50, it is a good idea to set the TextBoxSize (see page 547) property to the same number.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(50)]
[Description("The maximum number of characters the user can enter in the search box.")]
public int MaxCharacters;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(50)>
<Description("The maximum number of characters the user can enter in the search box.")>
Public Property MaxCharacters() As Integer
```

WebSearch.MaxTeaserLength Property

An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary is truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT).

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(256)]
[Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")]
public int MaxTeaserLength;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(256)>
<Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")>
Public Property MaxTeaserLength() As Integer
```

WebSearch.MultipleFolders Property

Optional: If you want to display an image on the submit button, enter the server path to it.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Optional: Multiple CMS Folders to search, seperate each folder with comma.")]
public string MultipleFolders;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Optional: Multiple CMS Folders to search, seperate each folder with comma.")>
Public Property MultipleFolders() As string
```

WebSearch.OrderBy Property

The order of the search results. Choices are:

- **Title** - The content title (alphabetical)
- **ID** - The content ID number
- **Date Created** - The date the content was created
- **Date Modified** - The date the content was most recently modified
- **Editor** - The user who last edited the content (alphabetical)
- **Rank** - The rank assigned to the content. See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 725)" > "Display of Search (see page 474) Results" > "Search (see page 474) Result Ranking"

C#

```
[Browsable(true)]
[Category("Ektron")]
[Description("Specify the order of the results. Choices are: Rank, Title, User, DateCreated,
DateModified")]
[Personalizable(), WebBrowsable()]
public WebSearchOrder OrderBy;
```

Visual Basic

```
<Browsable(true)>
<Category("Ektron")>
<Description("Specify the order of the results. Choices are: Rank, Title, User, DateCreated,
DateModified")>
<Personalizable(), WebBrowsable()>
Public Property OrderBy() As WebSearchOrder
```

WebSearch.OrderDirection Property

Specify the order direction of the results. Choices are **Ascending** or **Descending**.

C#

```
[Browsable(true)]
[Category("Ektron")]
[TypeConverter(typeof(SortOrderTypeConverter))]
[Description("Specify the order direction of the results.")]
[Personalizable(), WebBrowsable()]
public string OrderDirection;
```

Visual Basic

```
<Browsable(true)>
<Category("Ektron")>
<TypeConverter(typeof(SortOrderTypeConverter))>
<Description("Specify the order direction of the results.")>
<Personalizable(), WebBrowsable()>
Public Property OrderDirection() As string
```

WebSearch.Recursive Property

Select **True** if you wish to search sub-folders of the starting folder. The starting folder is identified in the `FolderID` (see page 536) property.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Select true if you wish to search the subfolders of the selected Folder")]
public bool Recursive;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Select true if you wish to search the subfolders of the selected Folder")>
Public Property Recursive() As bool
```

WebSearch.RemoveTeaserHtml Property

An optional result formatting parameter. If set to **True**, any HTML in the content items' teaser/summary is removed before it is sent to the browser. If set to **False**, any existing HTML is passed along, (this may then be further filtered by the XSLT).

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("An optional search-result formatting parameter that removes all HTML from each
content-items' teaser/summary before sending it to the browser.")]
public bool RemoveTeaserHtml;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("An optional search-result formatting parameter that removes all HTML from each
content-items' teaser/summary before sending it to the browser.")>
Public Property RemoveTeaserHtml() As bool
```

WebSearch.ResultCount Property

The number of search results. This is a read-only property.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("The number of search results. This is a read-only property.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int ResultCount;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The number of search results. This is a read-only property.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property ResultCount() As Integer
```

WebSearch.ResultDataTable Property

Allows a developer to get results from the search control as a DataTable

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to get results from the search control as a DataTable.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public virtual DataTable ResultDataTable;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows a developer to get results from the search control as a DataTable.")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Browsable(false)>
Public virtual Property ResultDataTable() As DataTable
```

WebSearch.ResultsPageSize Property

An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary is truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT).

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("An optional page size setting, defaults to web.config search provider page
size).")]
[Personalizable(), WebBrowsable()]
public int ResultsPageSize;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("An optional page size setting, defaults to web.config search provider page
```

```
size).")>
<Personalizable(), WebBrowsable()>
Public Property ResultsPageSize() As Integer
```

WebSearch.ResultTagId Property

Lets you designate where search results appear. You can place search criteria in one area of a Web form and the results in another.

For example, you have the following tag.

```
<span id="results"></span>
```

In this case, enter **results** for the property value.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("an optional parameter to specify the div control id to display the search
result")]
public string ResultTagId;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("an optional parameter to specify the div control id to display the search
result")>
Public Property ResultTagId() As string
```

WebSearch.ResultXml Property

Allows a developer to get results from the search control as XML.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to get results from the search control as XML.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public virtual string ResultXml;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows a developer to get results from the search control as XML.")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Browsable(false)>
Public virtual Property ResultXml() As string
```

WebSearch.SearchFor Property

Choose the type of content to search.

- All
- HTML
- Documents
- Images

- Multimedia
- Discussion Forums
- Tags (see page 805)

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Choose the type of content to search. All - HTML - Documents - Images -
Multimedia - Discussion Forums - Tags")]
[Personalizable(), WebBrowsable()]
public SearchDocumentType SearchFor;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Choose the type of content to search. All - HTML - Documents - Images -
Multimedia - Discussion Forums - Tags")>
<Personalizable(), WebBrowsable()>
Public Property SearchFor() As SearchDocumentType
```

WebSearch.SearchSynonyms Property

If set to **True**, the search results will search for synonyms configured for term. The default is True.

- **True** - Display search for synonym of term specified.
- **False** - Do not search synonyms.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to true, Synonyms configured will be used for search.")]
[Personalizable(), WebBrowsable()]
public bool SearchSynonyms;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to true, Synonyms configured will be used for search.")>
<Personalizable(), WebBrowsable()>
Public Property SearchSynonyms() As bool
```

WebSearch.SearchText Property

Get or set default search text. This is optional.

C#

```
[Browsable(false)]
public string SearchText;
```

Visual Basic

```
<Browsable(false)>
Public Property SearchText() As string
```

WebSearch.ShowCategories Property

If set to **True**, taxonomy categories will also be searched for the search text specified. The default is False.

True = Search (see page 474) across taxonomy categories

False = Do not search across taxonomy categories

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If this property is set to true, taxonomy categories will also be searched for
the search text specified.")]
public bool ShowCategories;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If this property is set to true, taxonomy categories will also be searched for
the search text specified.")>
Public Property ShowCategories() As bool
```

WebSearch.ShowCustomSummary Property

If set to **True**, custom summary is used in search results. The default is False.

True = Display custom summary in search results.

False = Display characterization generated by indexing service in search results.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If this property is set to true, custom summary is displayed in search
results.")]
public bool ShowCustomSummary;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If this property is set to true, custom summary is displayed in search
results.")>
Public Property ShowCustomSummary() As bool
```

WebSearch.ShowSearchBoxAlways Property

If set to **False**, the search box does not appear on PostBack. The default is True.

True = Display search box on PostBack

False = Do not display search box on PostBack

C#

```
[Bindable(true)]
[Browsable(true)]
```

```
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to false, the search box will not be visible on
PostBack.")]
public bool ShowSearchBoxAlways;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to false, the search box will not be visible on
PostBack.")>
Public Property ShowSearchBoxAlways() As bool
```

WebSearch.ShowSearchOptions Property

If set to **True**, the following options appear above the Search (see page 474) box.

- Site (see page 725)
- Html
- Documents
- Images
- Multimedia
- Tags (see page 805)

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to show these options above the Search box: Site, Html, Documents,
Images and Multimeadia.")]
public bool ShowSearchOptions;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to show these options above the Search box: Site, Html, Documents,
Images and Multimeadia.")>
Public Property ShowSearchOptions() As bool
```

WebSearch.ShowSuggestedResults Property

If set to **True**, the search results include Suggested Results if any exist. The default is True.

- **True** = Display Suggested Results
- **False** = Do not display Suggested Results

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to true, Suggested Results will be shown if any exist
for the search terms the user entered.")]
[Personalizable(), WebBrowsable()]
public bool ShowSuggestedResults;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to true, Suggested Results will be shown if any exist
for the search terms the user entered.")>
<Personalizable(), WebBrowsable()>
Public Property ShowSuggestedResults() As bool
```

WebSearch.StartCount Property

Get the numerical record of the last item on a page.

C#

```
[Browsable(false)]
public int StartCount;
```

Visual Basic

```
<Browsable(false)>
Public Property StartCount() As Integer
```

WebSearch.Stylesheet Property

Specify the location of the style sheet for the search results page. Set the location relative to the site root folder. For example: `csslib\mytest.css`.

Leave blank to use the default style sheet,

C#

```
[Description("Specify the style sheet relative to site root. Leave blank for default.")]
[Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(StringConverter))]
public string Stylesheet;
```

Visual Basic

```
<Description("Specify the style sheet relative to site root. Leave blank for default.")>
<Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(StringConverter))>
Public Property Stylesheet() As string
```

WebSearch.TaxCategories Property

Is server control in workarea

C#

```
[Browsable(false)]
public string TaxCategories;
```

Visual Basic

```
<Browsable(false)>
Public Property TaxCategories() As string
```

WebSearch.TaxonomyOperator Property

Taxonomy (see page 876) operator used in filtering search results by taxonomy category.

- and - selected taxonomy categories are AND-ed

- or - selected taxonomy categories are OR-ed

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Taxonomy operator used in filtering search results by taxonomy category.")]
[Personalizable(), WebBrowsable()]
public TaxCategoryOperator TaxonomyOperator;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Taxonomy operator used in filtering search results by taxonomy category.")>
<Personalizable(), WebBrowsable()>
Public Property TaxonomyOperator() As TaxCategoryOperator
```

WebSearch.TextBoxSize Property

The size of the Search (see page 474) text box for user input. The width is based on the number of characters.

C#

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The width of the search box, in characters.")]
public int TextBoxSize;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The width of the search box, in characters.")>
Public Property TextBoxSize() As Integer
```

WebSearch.TimeTaken Property

Gets the amount of time, in seconds, it has taken to perform the search.

C#

```
[Browsable(false)]
public double TimeTaken;
```

Visual Basic

```
<Browsable(false)>
Public Property TimeTaken() As double
```

WebSearch.XmlDoc Property

This property is not supported by WebSearch (see page 529). The underlying XML can be displayed using an XSLT specified by the DisplayXslt (see page 535) property of the WebSearch (see page 529) server control. The XSLT filename is specified relative to the current directory.

C#

```
[Bindable(true)]
[Browsable(false)]
[Description("This property is not supported by WebSearch. To retrieve the XML structure, use
the DisplayXslt property in the WebSearch server control to get search results formatted using
```

```
an XSLT. You need to either specify an XSLT name or use the default XSLTs (ecmNavigation,
ecmTeaser) provided with the server control.")]
public override XmlDocument XmlDoc;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Description("This property is not supported by WebSearch. To retrieve the XML structure, use
the DisplayXslt property in the WebSearch server control to get search results formatted using
an XSLT. You need to either specify an XSLT name or use the default XSLTs (ecmNavigation,
ecmTeaser) provided with the server control.")>
Public override Property XmlDoc() As XmlDocument
```

Structs, Records, Enums

The following table lists structs, records, enums in this documentation.

Enumerations

	Name	Description
	CommunityDocumentsObjectType (see page 548)	This control allows user / groups to manage their CommunityDocuments (see page 106).

Ektron.Cms.Controls.CommunityDocumentsObjectType Enumeration

This control allows user / groups to manage their CommunityDocuments (see page 106).

C#

```
public enum CommunityDocumentsObjectType {
}
```

Visual Basic

```
Public Enum CommunityDocumentsObjectType
End Enum
```

File

communitydocuments.cs

Ektron.Cms.Controls.WebCalendarForms Namespace

This is namespace Ektron.Cms.Controls.WebCalendarForms.

Classes

	Name	Description
	AdvancedEditTemplate (see page 549)	This is class Ektron.Cms.Controls.WebCalendarForms.AdvancedEditTemplate.
	AdvancedFormBase (see page 549)	This is class Ektron.Cms.Controls.WebCalendarForms.AdvancedFormBase.
	AdvancedInsertTemplate (see page 550)	This is class Ektron.Cms.Controls.WebCalendarForms.AdvancedInsertTemplate.
	AdvancedTemplate (see page 550)	This is class Ektron.Cms.Controls.WebCalendarForms.AdvancedTemplate.
	AppointmentTemplate (see page 550)	This is class Ektron.Cms.Controls.WebCalendarForms.AppointmentTemplate.
	DataKeys (see page 551)	This is class Ektron.Cms.Controls.WebCalendarForms.DataKeys.
	DisplayTemplate (see page 551)	This is class Ektron.Cms.Controls.WebCalendarForms.DisplayTemplate.
	SchedulerStyles (see page 551)	This is class Ektron.Cms.Controls.WebCalendarForms.SchedulerStyles.

Classes

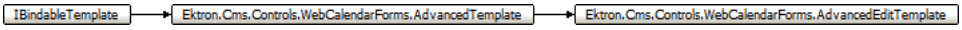
The following table lists classes in this documentation.

Classes

	Name	Description
	AdvancedEditTemplate (see page 549)	This is class Ektron.Cms.Controls.WebCalendarForms.AdvancedEditTemplate.
	AdvancedFormBase (see page 549)	This is class Ektron.Cms.Controls.WebCalendarForms.AdvancedFormBase.
	AdvancedInsertTemplate (see page 550)	This is class Ektron.Cms.Controls.WebCalendarForms.AdvancedInsertTemplate.
	AdvancedTemplate (see page 550)	This is class Ektron.Cms.Controls.WebCalendarForms.AdvancedTemplate.
	AppointmentTemplate (see page 550)	This is class Ektron.Cms.Controls.WebCalendarForms.AppointmentTemplate.
	DataKeys (see page 551)	This is class Ektron.Cms.Controls.WebCalendarForms.DataKeys.
	DisplayTemplate (see page 551)	This is class Ektron.Cms.Controls.WebCalendarForms.DisplayTemplate.
	SchedulerStyles (see page 551)	This is class Ektron.Cms.Controls.WebCalendarForms.SchedulerStyles.

AdvancedEditTemplate Class

Class Hierarchy



C#

```
internal class AdvancedEditTemplate : AdvancedTemplate;
```

Visual Basic

```
Friend Class AdvancedEditTemplate
Inherits AdvancedTemplate
```

File

AdvancedTemplate.cs

Description

This is class Ektron.Cms.Controls.WebCalendarForms.AdvancedEditTemplate.

AdvancedFormBase Class

Class Hierarchy

C#

```
public abstract class AdvancedFormBase : UserControl;
```

Visual Basic

```
Public abstract Class AdvancedFormBase
Inherits UserControl
```

File

AdvancedTemplate.cs

Description

This is class Ektron.Cms.Controls.WebCalendarForms.AdvancedFormBase.

AdvancedInsertTemplate Class

Class Hierarchy

C#

```
internal class AdvancedInsertTemplate : AdvancedTemplate;
```

Visual Basic

```
Friend Class AdvancedInsertTemplate
Inherits AdvancedTemplate
```

File

AdvancedTemplate.cs

Description

This is class Ektron.Cms.Controls.WebCalendarForms.AdvancedInsertTemplate.

AdvancedTemplate Class

Class Hierarchy

C#

```
internal abstract class AdvancedTemplate : IBindableTemplate;
```

Visual Basic

```
Friend abstract Class AdvancedTemplate
Inherits IBindableTemplate
```

File

AdvancedTemplate.cs

Description

This is class Ektron.Cms.Controls.WebCalendarForms.AdvancedTemplate.

AppointmentTemplate Class

Class Hierarchy

C#

```
internal class AppointmentTemplate : ITemplate;
```

Visual Basic

```
Friend Class AppointmentTemplate
Inherits ITemplate
```

File

DisplayTemplate.cs

Description

This is class Ektron.Cms.Controls.WebCalendarForms.AppointmentTemplate.

DataKeys Class

Class Hierarchy

Ektron.Cms.Controls.WebCalendarForms.DataKeys

C#

```
internal static class DataKeys;
```

Visual Basic

```
Friend static Class DataKeys
```

File

AdvancedTemplate.cs

Description

This is class Ektron.Cms.Controls.WebCalendarForms.DataKeys.

DisplayTemplate Class

Class Hierarchy

UserControl → Ektron.Cms.Controls.WebCalendarForms.DisplayTemplate

C#

```
public abstract class DisplayTemplate : UserControl;
```

Visual Basic

```
Public abstract Class DisplayTemplate
Inherits UserControl
```

File

DisplayTemplate.cs

Description

This is class Ektron.Cms.Controls.WebCalendarForms.DisplayTemplate.

SchedulerStyles Class

Class Hierarchy

Ektron.Cms.Controls.WebCalendarForms.SchedulerStyles

C#

```
internal class SchedulerStyles;
```

Visual Basic

```
Friend Class SchedulerStyles
```

File

AdvancedTemplate.cs

Description

This is class Ektron.Cms.Controls.WebCalendarForms.SchedulerStyles.

Plug-in Extension API

The Plug-in Extension API documentation details the methods and properties that are available when creating a custom plug-in.

Plug-in Extensions expose event hooks in Ektron CMS400.NET. As a developer, you can utilize these event hooks in Visual Studio C# or VB.NET to create customized events for your site. For instance, you could create a Plug-in Extension that:

- sends an instant message when content is published
- sends an email when content is published
- filters out objectionable language from blog posts
- appends HTML content when it's published

For additional information on Plug-in Extensions, see the Developer Manual section "Plug-in Extensions"

Namespaces

Name	Description
Extensibility (see page 552)	This is namespace Ektron.Cms.Extensibility.

Ektron.Cms.Extensibility Namespace

This is namespace Ektron.Cms.Extensibility.

Classes

	Name	Description
	ActionRecipientInfo (see page 553)	Class which holds information about the Invitation Recipient being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	ActionRequestInfo (see page 555)	Class which holds information about the Invitation being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	ContentInfo (see page 558)	Class which holds information about the content being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: long = -1. string = "". bool = false.
	ExtensionAuthor (see page 561)	Custom attribute - Used at the method and/or assembly level to indicate who wrote the plug-in.
	ExtensionDescription (see page 562)	Custom attribute - Used at the method level to describe what a particular event does
	ExtensionEvent (see page 563)	Override the methods in this class to customize the CMS via event hooks.
	FlagInfo (see page 581)	Class which holds information about the flagged content. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	FolderInfo (see page 585)	Class which holds information about the folder being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	FormField (see page 587)	Class that describes a field in a form - name / value pair
	FormInfo (see page 588)	Class which describes a Form (see page 871) object

	MessageBoardInfo (see page 590)	Class which holds information about the Invitation Recipient being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	RequestInfo (see page 591)	Reserved for future use.
	ReviewInfo (see page 591)	Class which holds information about the review being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	SiteInfo (see page 595)	Class which holds information about the site in which the event was fired.
	UserInfo (see page 596)	Class which holds information about the user being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	ActionRecipientInfo (see page 553)	Class which holds information about the Invitation Recipient being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	ActionRequestInfo (see page 555)	Class which holds information about the Invitation being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	ContentInfo (see page 558)	Class which holds information about the content being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: long = -1. string = "". bool = false.
	ExtensionAuthor (see page 561)	Custom attribute - Used at the method and/or assembly level to indicate who wrote the plug-in.
	ExtensionDescription (see page 562)	Custom attribute - Used at the method level to describe what a particular event does
	ExtensionEvent (see page 563)	Override the methods in this class to customize the CMS via event hooks.
	FlagInfo (see page 581)	Class which holds information about the flagged content. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	FolderInfo (see page 585)	Class which holds information about the folder being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	FormField (see page 587)	Class that describes a field in a form - name / value pair
	FormInfo (see page 588)	Class which describes a Form (see page 871) object
	MessageBoardInfo (see page 590)	Class which holds information about the Invitation Recipient being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	RequestInfo (see page 591)	Reserved for future use.
	ReviewInfo (see page 591)	Class which holds information about the review being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	SiteInfo (see page 595)	Class which holds information about the site in which the event was fired.
	UserInfo (see page 596)	Class which holds information about the user being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

ActionRecipientInfo Class

Class which holds information about the Invitation Recipient being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

Class Hierarchy

Ektron.Cms.Extensibility.ActionRecipientInfo

C#

```
public sealed class ActionRecipientInfo;
```

Visual Basic

```
Public sealed Class ActionRecipientInfo
```

File

CmsInfoObjects.cs

ActionRecipientInfo.ActionRecipientInfo Constructor

Constructor

C#

```
public ActionRecipientInfo(XmlNode xml);
```

Visual Basic

```
Public Sub New(xml As XmlNode)
```

Parameters

Parameters	Description
XmlNode xml	node containing Recipient info.

ActionRecipientInfo Properties

The properties of the ActionRecipientInfo class are listed here.

Public Properties

	Name	Description
	DisplayName (see page 554)	Display Name of the recipient.
	Email (see page 554)	Email address of the recipient
	FirstName (see page 555)	First Name of the recipient.
	IsUser (see page 555)	Returns true if the recipient is a validated user. This call is only valid after the Invitation has been Validated; otherwise false is returned regardless.
	LastName (see page 555)	Last Name of the recipient.
	UserId (see page 555)	a numeric value representing the CMS User (see page 950) ID of the recipient.

ActionRecipientInfo.DisplayName Property

Display Name of the recipient.

C#

```
public string DisplayName;
```

Visual Basic

```
Public Property DisplayName() As string
```

ActionRecipientInfo.Email Property

Email address of the recipient

C#

```
public string Email;
```

Visual Basic

```
Public Property Email() As string
```

ActionRecipientInfo.FirstName Property

First Name of the recipient.

C#

```
public string FirstName;
```

Visual Basic

```
Public Property FirstName() As string
```

ActionRecipientInfo.IsUser Property

Returns true if the recipient is a validated user. This call is only valid after the Invitation has been Validated; otherwise false is returned regardless.

C#

```
public bool IsUser;
```

Visual Basic

```
Public Property IsUser() As bool
```

ActionRecipientInfo.LastName Property

Last Name of the recipient.

C#

```
public string LastName;
```

Visual Basic

```
Public Property LastName() As string
```

ActionRecipientInfo.UserId Property

a numeric value representing the CMS User (see page 950) ID of the recipient.

C#

```
public long UserId;
```

Visual Basic

```
Public Property UserId() As long
```

ActionRequestInfo Class

Class which holds information about the Invitation being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

Class Hierarchy

Ektron.Cms.Extensibility.ActionRequestInfo

C#

```
public sealed class ActionRequestInfo;
```

Visual Basic

```
Public sealed Class ActionRequestInfo
```

File

CmsInfoObjects.cs

ActionRequestInfo.ActionRequestInfo Constructor

C#

```
public ActionRequestInfo(XmlNode xml);
```

Visual Basic

```
Public Sub New(xml As XmlNode)
```

Parameters

Parameters	Description
XmlNode xml	

ActionRequestInfo Enumerations

The enumerations of the ActionRequestInfo class are listed here.

Enumerations

	Name	Description
	ActionRequestType (see page 556)	Enumeration identifying the different types of ActionRequests.

Ektron.Cms.Extensibility.ActionRequestInfo.ActionRequestType Enumeration

Enumeration identifying the different types of ActionRequests.

C#

```
public enum ActionRequestType {
  Friend = 1,
  JoinGroup = 2
}
```

Visual Basic

```
Public Enum ActionRequestType
  Friend = 1
  JoinGroup = 2
End Enum
```

File

CmsInfoObjects.cs

Members

Members	Description
Friend = 1	Represents a friend invitation.
JoinGroup = 2	Represents a request to join a prviate group.

ActionRequestInfo Properties

The properties of the ActionRequestInfo class are listed here.

Public Properties

	Name	Description
	ErrorMessage (see page 557)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	Id (see page 557)	A numeric value representing the ID of the invitation. Only populated after save.
	Recipients (see page 557)	Collection (see page 95) recipients for the invitation
	SenderId (see page 557)	A numeric value representing the user ID of the invitation sender.
	Type (see page 557)	User (see page 950) ID of the invitation sender.

ActionRequestInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

C#

```
public string ErrorMessage;
```

Visual Basic

```
Public Property ErrorMessage() As string
```

ActionRequestInfo.Id Property

A numeric value representing the ID of the invitation. Only populated after save.

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

ActionRequestInfo.Recipients Property

Collection (see page 95) recipients for the invitation

C#

```
public Collection<ActionRecipientInfo> Recipients;
```

Visual Basic

```
Public Property Recipients() As Collection<ActionRecipientInfo>
```

ActionRequestInfo.SenderId Property

A numeric value representing the user ID of the invitation sender.

C#

```
public long SenderId;
```

Visual Basic

```
Public Property SenderId() As long
```

ActionRequestInfo.Type Property

User (see page 950) ID of the invitation sender.

C#

```
public ActionRequestType Type;
```

Visual Basic

```
Public Property Type() As ActionRequestType
```

ContentInfo Class

Class which holds information about the content being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: long = -1. string = "". bool = false.

Class Hierarchy

Ektron.Cms.Extensibility.ContentInfo

C#

```
public sealed class ContentInfo;
```

Visual Basic

```
Public sealed Class ContentInfo
```

File

CmsInfoObjects.cs

ContentInfo Properties

The properties of the ContentInfo class are listed here.

Public Properties

	Name	Description
	AssetID (see page 558)	Gets a string containing the asset ID (see page 560) in the form of a GUID if the content is an asset. (ReadOnly)
	Comments (see page 559)	Gets a string containing the comments a user has entered during their operation. (ReadOnly)
	ErrorMessage (see page 559)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	FileAndPath (see page 559)	Gets a string containing the full physical path and file name on the server if content is an asset. (ReadOnly)
	FileExtension (see page 559)	Gets a string containing file extension if content is an asset. (ReadOnly)
	FileName (see page 559)	Gets a string containing file name if content is an asset. (ReadOnly)
	FolderID (see page 559)	Gets an long representing the folder ID (see page 560) which this content is located. (ReadOnly)
	Html (see page 560)	Gets or Sets a string containing all the html displayed to end users for a piece of content. (ReadWrite Note: This property only allows write access within OnBeforePublish.)
	ID (see page 560)	Gets an long representing the content ID of the content. (ReadOnly)
	Language (see page 560)	Gets an integer representing the language of the content. (ReadOnly)
	MimeType (see page 560)	Gets a string containing the MIME type if content is an asset. (ReadOnly)
	Status (see page 560)	Gets a string representing the status of the content, published, checkedOut. (ReadOnly)
	Title (see page 560)	Gets a string containing the title of the content. (ReadOnly)
	Type (see page 561)	Gets an integer representing the content type. (ReadOnly)
	UserID (see page 561)	Gets a long representing the user ID (see page 560) of the user performing the operation on the content. (ReadOnly)

ContentInfo.AssetID Property

Gets a string containing the asset ID (see page 560) in the form of a GUID if the content is an asset. (ReadOnly)

C#

```
public string AssetID;
```

Visual Basic

```
Public Property AssetID() As string
```

ContentInfo.Comments Property

Gets a string containing the comments a user has entered during their operation. (ReadOnly)

C#

```
public string Comments;
```

Visual Basic

```
Public Property Comments() As string
```

ContentInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

C#

```
public string ErrorMessage;
```

Visual Basic

```
Public Property ErrorMessage() As string
```

ContentInfo.FileAndPath Property

Gets a string containing the full physical path and file name on the server if content is an asset. (ReadOnly)

C#

```
public string FileAndPath;
```

Visual Basic

```
Public Property FileAndPath() As string
```

ContentInfo.FileExtension Property

Gets a string containing file extension if content is an asset. (ReadOnly)

C#

```
public string FileExtension;
```

Visual Basic

```
Public Property FileExtension() As string
```

ContentInfo.FileName Property

Gets a string containing file name if content is an asset. (ReadOnly)

C#

```
public string FileName;
```

Visual Basic

```
Public Property FileName() As string
```

ContentInfo.FolderID Property

Gets an long representing the folder ID (see page 560) which this content is located. (ReadOnly)

C#

```
public long FolderID;
```

Visual Basic

```
Public Property FolderID() As long
```

ContentInfo.Html Property

Gets or Sets a string containing all the html displayed to end users for a piece of content. (ReadWrite Note: This property only allows write access within OnBeforePublish.)

C#

```
public string Html;
```

Visual Basic

```
Public Property Html() As string
```

ContentInfo.ID Property

Gets an long representing the content ID of the content. (ReadOnly)

C#

```
public long ID;
```

Visual Basic

```
Public Property ID() As long
```

ContentInfo.Language Property

Gets an integer representing the language of the content. (ReadOnly)

C#

```
public int Language;
```

Visual Basic

```
Public Property Language() As Integer
```

ContentInfo.MimeType Property

Gets a string containing the MIME type if content is an asset. (ReadOnly)

C#

```
public string MimeType;
```

Visual Basic

```
Public Property MimeType() As string
```

ContentInfo.Status Property

Gets a string representing the status of the content, published, checkedOut. (ReadOnly)

C#

```
public string Status;
```

Visual Basic

```
Public Property Status() As string
```

ContentInfo.Title Property

Gets a string containing the title of the content. (ReadOnly)

C#

```
public string Title;
```

Visual Basic

```
Public Property Title() As string
```

ContentInfo.Type Property

Gets an integer representing the content type. (ReadOnly)

C#

```
public int Type;
```

Visual Basic

```
Public Property Type() As Integer
```

ContentInfo.UserID Property

Gets a long representing the user ID (see page 560) of the user performing the operation on the content. (ReadOnly)

C#

```
public long UserID;
```

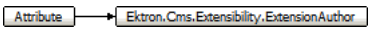
Visual Basic

```
Public Property UserID() As long
```

ExtensionAuthor Class

Custom attribute - Used at the method and/or assembly level to indicate who wrote the plug-in.

Class Hierarchy



C#

```
[AttributeUsage(AttributeTargets.Method | AttributeTargets.Assembly)]
public class ExtensionAuthor : Attribute;
```

Visual Basic

```
<AttributeUsage(AttributeTargets.Method | AttributeTargets.Assembly)>
Public Class ExtensionAuthor
Inherits Attribute
```

File

ExtensibilityBase.cs

ExtensionAuthor.ExtensionAuthor Constructor

Sets the creator of the Extension.

C#

```
public ExtensionAuthor(string author);
```

Visual Basic

```
Public Sub New(author As string)
```

Parameters

Parameters	Description
string author	String with name of author who created Extension.

ExtensionAuthor Methods

The methods of the ExtensionAuthor class are listed here.

Public Methods

	Name	Description
	ToString (see page 562)	Converts author name to a string.

ExtensionAuthor.ToString Method

Converts author name to a string.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Returns

String representing the author's name.

ExtensionAuthor Properties

The properties of the ExtensionAuthor class are listed here.

Public Properties

	Name	Description
	Author (see page 562)	Gets the creator of the Extension. (Read Only)

ExtensionAuthor.Author Property

Gets the creator of the Extension. (Read Only)

C#

```
public string Author;
```

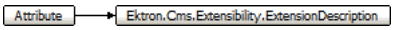
Visual Basic

```
Public Property Author() As string
```

ExtensionDescription Class

Custom attribute - Used at the method level to describe what a particular event does

Class Hierarchy



C#

```
[AttributeUsage(AttributeTargets.Method)]
public class ExtensionDescription : Attribute;
```

Visual Basic

```
<AttributeUsage(AttributeTargets.Method)>
Public Class ExtensionDescription
Inherits Attribute
```

File

ExtensibilityBase.cs

ExtensionDescription.ExtensionDescription Constructor

Sets the description of the Extension.

C#

```
public ExtensionDescription(string description);
```

Visual Basic

```
Public Sub New(description As string)
```

Parameters

Parameters	Description
string description	String representing the description of the Extension

ExtensionDescription Methods

The methods of the ExtensionDescription class are listed here.

Public Methods

	Name	Description
	ToString (see page 563)	Converts the description to a string.

ExtensionDescription.ToString Method

Converts the description to a string.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

Returns

String representing the description of the Extension.

ExtensionDescription Properties

The properties of the ExtensionDescription class are listed here.

Public Properties

	Name	Description
	Desc (see page 563)	Gets the description of the extension. (ReadOnly)

ExtensionDescription.Desc Property

Gets the description of the extension. (ReadOnly)

C#

```
public string Desc;
```

Visual Basic

```
Public Property Desc() As string
```

ExtensionEvent Class

Override the methods in this class to customize the CMS via event hooks.

Class Hierarchy

C#

```
[Serializable]
public abstract class ExtensionEvent : IWebServiceProxy;
```

Visual Basic

```
<Serializable>
Public abstract Class ExtensionEvent
Inherits IWebServiceProxy
```

File

ExtensibilityBase.cs

ExtensionEvent Methods

The methods of the ExtensionEvent class are listed here.

Public Methods

	Name	Description
	GetActiveDirectoryUserProxy (see page 566)	Returns an ActiveDirectoryUserProxy object to access the ActiveDirectoryUser (see page 1103) Web Service.
	GetAssetProxy (see page 566)	Returns an AssetProxy object to access the Asset (see page 822) Web Service.
	GetBlogProxy (see page 566)	Returns a BlogProxy object to access the Blog (see page 826) Web Service.
	GetBusinessRulesProxy (see page 567)	Returns a BusinessRulesProxy object to access the BusinessRules (see page 73) Web Service.
	GetCalendarEventTypeProxy (see page 567)	Returns a CalendarEventTypeProxy object to access the CalendarEventType Web Service.
	GetCalendarProxy (see page 567)	Returns a CalendarProxy object to access the Calendar (see page 76) Web Service.
	GetCommunityGroupProxy (see page 567)	Returns a CommunityGroupProxy object to access the CommunityGroup (see page 750) Web Service.
	GetContentProxy (see page 567)	Returns a ContentProxy object to access the Content (see page 580) Web Service.
	GetContentRatingProxy (see page 568)	Returns a ContentRatingProxy object to access the ContentRating (see page 866) Web Service.
	GetCustomFieldsProxy (see page 568)	Returns a CustomFieldsProxy object to access the CustomFields (see page 673) Web Service.
	GetFavoritesProxy (see page 568)	Returns a FavoritesProxy object to access the Favorites (see page 763) Web Service.
	GetFlagProxy (see page 568)	Returns a FlaggedDataProxy object to access the Content (see page 580) flagging Web Service.
	GetFolderProxy (see page 569)	Returns a FolderProxy object to access the Folder (see page 581) Web Service.
	GetFontProxy (see page 569)	Returns a FontProxy object to access the Font (see page 1244) Web Service.
	GetFormProxy (see page 569)	Returns a FormProxy object to access the Form (see page 581) Web Service.
	GetFriendsProxy (see page 569)	Returns a FriendsProxy object to access the FavoriFriendstes Web Service.
	GetLibraryProxy (see page 570)	Returns a LibraryProxy object to access the Library (see page 695) Web Service.
	GetMessageBoardProxy (see page 570)	Returns a MessageBoardProxy object to access the MessageBoard (see page 787) Web Service.
	GetMetadataProxy (see page 570)	Returns a MetadataProxy object to access the Metadata (see page 714) Web Service.

	GetObjectStateAsXml (see page 570)	Retrieves changes made by individual plug-ins
	GetPermissionsProxy (see page 570)	Returns a PermissionsProxy object to access the Permissions (see page 721) Web Service.
	GetSearchManagerProxy (see page 571)	Returns a SearchManagerProxy object to access the SearchManager (see page 899) Web Service.
	GetSiteMapProxy (see page 571)	Returns a SiteMapProxy object to access the SiteMap (see page 732) Web Service.
	GetSiteProxy (see page 571)	Returns a SiteProxy object to access the Site (see page 581) Web Service.
	GetTagsProxy (see page 571)	Returns a TagsProxy object to access the Tags (see page 805) Web Service.
	GetTaskCategoryProxy (see page 572)	Returns a TaskCategoryProxy object to access the Task (see page 909) Category Web Service.
	GetTaskCategoryTypeProxy (see page 572)	Returns a TaskProxy object to access the Task (see page 909) Web Service.
	GetTaskProxy (see page 572)	Returns a TaskProxy object to access the Task (see page 909) Web Service.
	GetTaxonomyProxy (see page 572)	Returns a TaxonomyDataProxy object to access the taxonomy Web Service.
	GetThreadedDiscussionProxy (see page 573)	Returns a ThreadedDiscussionProxy object to access the ThreadedDiscussion (see page 882) Web Service.
	GetUserProxy (see page 573)	Returns a UserProxy object to access the User (see page 581) Web Service.
	OnAfterActionRequestSent (see page 573)	This hook is called after an ActionRequest is Sent.
	OnAfterContentAdd (see page 573)	This hook is called after content is added in the CMS.
	OnAfterContentDelete (see page 574)	This hook is called after a delete of content is completed in the CMS.
	OnAfterContentEdit (see page 574)	This hook is called after an edit is completed in the CMS.
	OnAfterContentFlagged (see page 574)	This hook is called after an Content (see page 580) is flagged.
	OnAfterFolderAdd (see page 574)	This hook is called after a folder is created in the CMS.
	OnAfterFolderDelete (see page 574)	This hook is called after a folder is deleted in the CMS.
	OnAfterFolderEdit (see page 575)	This hook is called after a folder edit is completed in the CMS.
	OnAfterFormSubmit (see page 575)	This hook is called after form data is submitted to the CMS database.
	OnAfterPublish (see page 575)	This hook is called after a publish is completed in the CMS.
	OnAfterReviewRejected (see page 575)	This hook is called after a review is rejected.
	OnAfterUserAdd (see page 576)	This hook is called after a user is created in the CMS.
	OnAfterUserDelete (see page 576)	This hook is called after a user is deleted from the CMS.
	OnAfterUserEdit (see page 576)	This hook is called after a user is edited in the CMS.
	OnAfterUserLogin (see page 576)	This hook is called after a user logs into in the CMS.
	OnBeforeActionRequestSent (see page 577)	This hook is called before an ActionRequest is Sent.
	OnBeforeContentAdd (see page 577)	This hook is called before content is added to the CMS.
	OnBeforeContentDelete (see page 577)	This hook is called before a delete of content in the CMS.
	OnBeforeContentEdit (see page 577)	This hook is called before a content is edited in the CMS.
	OnBeforeContentFlagged (see page 578)	This hook is called before an Content (see page 580) is flagged.

	OnBeforeFolderAdd (see page 578)	This hook is called before a folder is created in the CMS.
	OnBeforeFolderDelete (see page 578)	This hook is called before a folder is deleted from the CMS.
	OnBeforeFolderEdit (see page 578)	This hook is called before a folder is edited in the CMS.
	OnBeforeFormSubmit (see page 578)	This hook is called before form data is submitted to the CMS database.
	OnBeforePublish (see page 579)	This hook is called before a publish takes place in the CMS.
	OnBeforeReviewRejected (see page 579)	This hook is called before a review is rejected.
	OnBeforeUserAdd (see page 579)	This hook is called before a user is created in the CMS.
	OnBeforeUserDelete (see page 579)	This hook is called before a user is delete from the CMS.
	OnBeforeUserEdit (see page 580)	This hook is called before a user is edited in the CMS.
	OnBeforeUserLogin (see page 580)	This hook is called before a user login is initiated in the CMS.

ExtensionEvent.GetActiveDirectoryUserProxy Method

Returns an ActiveDirectoryUserProxy object to access the ActiveDirectoryUser (see page 1103) Web Service.

C#

```
public ActiveDirectoryUserProxy GetActiveDirectoryUserProxy();
```

Visual Basic

```
Public Function GetActiveDirectoryUserProxy() As ActiveDirectoryUserProxy
```

Returns

ActiveDirectoryUserProxy

ExtensionEvent.GetAssetProxy Method

Returns an AssetProxy object to access the Asset (see page 822) Web Service.

C#

```
public AssetProxy GetAssetProxy();
```

Visual Basic

```
Public Function GetAssetProxy() As AssetProxy
```

Returns

AssetProxy

ExtensionEvent.GetBlogProxy Method

Returns a BlogProxy object to access the Blog (see page 826) Web Service.

C#

```
public BlogProxy GetBlogProxy();
```

Visual Basic

```
Public Function GetBlogProxy() As BlogProxy
```

Returns

BlogProxy

ExtensionEvent.GetBusinessRulesProxy Method

Returns a BusinessRulesProxy object to access the BusinessRules (see page 73) Web Service.

C#

```
public BusinessRulesProxy GetBusinessRulesProxy();
```

Visual Basic

```
Public Function GetBusinessRulesProxy() As BusinessRulesProxy
```

Returns

BusinessRulesProxy

ExtensionEvent.GetCalendarEventTypeProxy Method

Returns a CalendarEventTypeProxy object to access the CalendarEventType Web Service.

C#

```
public CalendarEventTypeProxy GetCalendarEventTypeProxy();
```

Visual Basic

```
Public Function GetCalendarEventTypeProxy() As CalendarEventTypeProxy
```

Returns

CalendarEventTypeProxy

ExtensionEvent.GetCalendarProxy Method

Returns a CalendarProxy object to access the Calendar (see page 76) Web Service.

C#

```
public CalendarProxy GetCalendarProxy();
```

Visual Basic

```
Public Function GetCalendarProxy() As CalendarProxy
```

Returns

CalendarProxy

ExtensionEvent.GetCommunityGroupProxy Method

Returns a CommunityGroupProxy object to access the CommunityGroup (see page 750) Web Service.

C#

```
public CommunityGroupProxy GetCommunityGroupProxy();
```

Visual Basic

```
Public Function GetCommunityGroupProxy() As CommunityGroupProxy
```

Returns

CommunityGroupProxy

ExtensionEvent.GetContentProxy Method

Returns a ContentProxy object to access the Content (see page 580) Web Service.

C#

```
public ContentProxy GetContentProxy();
```

Visual Basic

```
Public Function GetContentProxy() As ContentProxy
```

Returns

ContentProxy

ExtensionEvent.GetContentRatingProxy Method

Returns a ContentRatingProxy object to access the ContentRating (see page 866) Web Service.

C#

```
public ContentRatingProxy GetContentRatingProxy();
```

Visual Basic

```
Public Function GetContentRatingProxy() As ContentRatingProxy
```

Returns

ContentRatingProxy

ExtensionEvent.GetCustomFieldsProxy Method

Returns a CustomFieldsProxy object to access the CustomFields (see page 673) Web Service.

C#

```
public CustomFieldsProxy GetCustomFieldsProxy();
```

Visual Basic

```
Public Function GetCustomFieldsProxy() As CustomFieldsProxy
```

Returns

CustomFieldsProxy

ExtensionEvent.GetFavoritesProxy Method

Returns a FavoritesProxy object to access the Favorites (see page 763) Web Service.

C#

```
public FavoritesProxy GetFavoritesProxy();
```

Visual Basic

```
Public Function GetFavoritesProxy() As FavoritesProxy
```

Returns

FavoritesProxy

ExtensionEvent.GetFlagProxy Method

Returns a FlaggedDataProxy object to access the Content (see page 580) flagging Web Service.

C#

```
public FlagProxy GetFlagProxy();
```

Visual Basic

```
Public Function GetFlagProxy() As FlagProxy
```

Returns

FlagProxy

ExtensionEvent.GetFolderProxy Method

Returns a FolderProxy object to access the Folder (see page 581) Web Service.

C#

```
public FolderProxy GetFolderProxy();
```

Visual Basic

```
Public Function GetFolderProxy() As FolderProxy
```

Returns

FolderProxy

ExtensionEvent.GetFontProxy Method

Returns a FontProxy object to access the Font (see page 1244) Web Service.

C#

```
public FontProxy GetFontProxy();
```

Visual Basic

```
Public Function GetFontProxy() As FontProxy
```

Returns

FontProxy

ExtensionEvent.GetFormProxy Method

Returns a FormProxy object to access the Form (see page 581) Web Service.

C#

```
public FormProxy GetFormProxy();
```

Visual Basic

```
Public Function GetFormProxy() As FormProxy
```

Returns

FormProxy

ExtensionEvent.GetFriendsProxy Method

Returns a FriendsProxy object to access the FavoriFriendstes Web Service.

C#

```
public FriendsProxy GetFriendsProxy();
```

Visual Basic

```
Public Function GetFriendsProxy() As FriendsProxy
```

Returns

FriendsProxy

ExtensionEvent.GetLibraryProxy Method

Returns a LibraryProxy object to access the Library (see page 695) Web Service.

C#

```
public LibraryProxy GetLibraryProxy();
```

Visual Basic

```
Public Function GetLibraryProxy() As LibraryProxy
```

Returns

LibraryProxy

ExtensionEvent.GetMessageBoardProxy Method

Returns a MessageBoardProxy object to access the MessageBoard (see page 787) Web Service.

C#

```
public MessageBoardProxy GetMessageBoardProxy();
```

Visual Basic

```
Public Function GetMessageBoardProxy() As MessageBoardProxy
```

Returns

MessageBoardProxy

ExtensionEvent.GetMetadataProxy Method

Returns a MetadataProxy object to access the Metadata (see page 714) Web Service.

C#

```
public MetadataProxy GetMetadataProxy();
```

Visual Basic

```
Public Function GetMetadataProxy() As MetadataProxy
```

Returns

MetadataProxy

ExtensionEvent.GetObjectStateAsXml Method

Retrieves changes made by individual plug-ins

C#

```
public string GetObjectStateAsXml();
```

Visual Basic

```
Public Function GetObjectStateAsXml() As string
```

ExtensionEvent.GetPermissionsProxy Method

Returns a PermissionsProxy object to access the Permissions (see page 721) Web Service.

C#

```
public PermissionsProxy GetPermissionsProxy();
```

Visual Basic

```
Public Function GetPermissionsProxy() As PermissionsProxy
```

Returns

PermissionsProxy

ExtensionEvent.GetSearchManagerProxy Method

Returns a SearchManagerProxy object to access the SearchManager (see page 899) Web Service.

C#

```
public SearchManagerProxy GetSearchManagerProxy();
```

Visual Basic

```
Public Function GetSearchManagerProxy() As SearchManagerProxy
```

Returns

SearchManagerProxy

ExtensionEvent.GetSiteMapProxy Method

Returns a SiteMapProxy object to access the SiteMap (see page 732) Web Service.

C#

```
public SiteMapProxy GetSiteMapProxy();
```

Visual Basic

```
Public Function GetSiteMapProxy() As SiteMapProxy
```

Returns

SiteMapProxy

ExtensionEvent.GetSiteProxy Method

Returns a SiteProxy object to access the Site (see page 581) Web Service.

C#

```
public SiteProxy GetSiteProxy();
```

Visual Basic

```
Public Function GetSiteProxy() As SiteProxy
```

Returns

SiteProxy

ExtensionEvent.GetTagsProxy Method

Returns a TagsProxy object to access the Tags (see page 805) Web Service.

C#

```
public TagsProxy GetTagsProxy();
```

Visual Basic

```
Public Function GetTagsProxy() As TagsProxy
```

Returns

TagsProxy

ExtensionEvent.GetTaskCategoryProxy Method

Returns a TaskCategoryProxy object to access the Task (see page 909) Category Web Service.

C#

```
public TaskCategoryProxy GetTaskCategoryProxy();
```

Visual Basic

```
Public Function GetTaskCategoryProxy() As TaskCategoryProxy
```

Returns

TaskCategoryProxy

ExtensionEvent.GetTaskCategoryTypeProxy Method

Returns a TaskProxy object to access the Task (see page 909) Web Service.

C#

```
public TaskCategoryTypeProxy GetTaskCategoryTypeProxy();
```

Visual Basic

```
Public Function GetTaskCategoryTypeProxy() As TaskCategoryTypeProxy
```

Returns

TaskProxy

ExtensionEvent.GetTaskProxy Method

Returns a TaskProxy object to access the Task (see page 909) Web Service.

C#

```
public TaskProxy GetTaskProxy();
```

Visual Basic

```
Public Function GetTaskProxy() As TaskProxy
```

Returns

TaskProxy

ExtensionEvent.GetTaxonomyProxy Method

Returns a TaxonomyDataProxy object to access the taxonomy Web Service.

C#

```
public TaxonomyProxy GetTaxonomyProxy();
```

Visual Basic

```
Public Function GetTaxonomyProxy() As TaxonomyProxy
```

Returns

TaxonomyProxy

ExtensionEvent.GetThreadedDiscussionProxy Method

Returns a ThreadedDiscussionProxy object to access the ThreadedDiscussion (see page 882) Web Service.

C#

```
public ThreadedDiscussionProxy GetThreadedDiscussionProxy();
```

Visual Basic

```
Public Function GetThreadedDiscussionProxy() As ThreadedDiscussionProxy
```

Returns

ThreadedDiscussionProxy

ExtensionEvent.GetUserProxy Method

Returns a UserProxy object to access the User (see page 581) Web Service.

C#

```
public UserProxy GetUserProxy();
```

Visual Basic

```
Public Function GetUserProxy() As UserProxy
```

Returns

UserProxy

ExtensionEvent.OnAfterActionRequestSent Method

This hook is called after an ActionRequest is Sent.

C#

```
public virtual bool OnAfterActionRequestSent();
```

Visual Basic

```
Public virtual Function OnAfterActionRequestSent() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterContentAdd Method

This hook is called after content is added in the CMS.

C#

```
public virtual bool OnAfterContentAdd();
```

Visual Basic

```
Public virtual Function OnAfterContentAdd() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterContentDelete Method

This hook is called after a delete of content is completed in the CMS.

C#

```
public virtual bool OnAfterContentDelete();
```

Visual Basic

```
Public virtual Function OnAfterContentDelete() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterContentEdit Method

This hook is called after an edit is completed in the CMS.

C#

```
public virtual bool OnAfterContentEdit();
```

Visual Basic

```
Public virtual Function OnAfterContentEdit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterContentFlagged Method

This hook is called after an Content (see page 580) is flagged.

C#

```
public virtual bool OnAfterContentFlagged();
```

Visual Basic

```
Public virtual Function OnAfterContentFlagged() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterFolderAdd Method

This hook is called after a folder is created in the CMS.

C#

```
public virtual bool OnAfterFolderAdd();
```

Visual Basic

```
Public virtual Function OnAfterFolderAdd() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterFolderDelete Method

This hook is called after a folder is deleted in the CMS.

C#

```
public virtual bool OnAfterFolderDelete();
```

Visual Basic

```
Public virtual Function OnAfterFolderDelete() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterFolderEdit Method

This hook is called after a folder edit is completed in the CMS.

C#

```
public virtual bool OnAfterFolderEdit();
```

Visual Basic

```
Public virtual Function OnAfterFolderEdit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterFormSubmit Method

This hook is called after form data is submitted to the CMS database.

C#

```
public virtual bool OnAfterFormSubmit();
```

Visual Basic

```
Public virtual Function OnAfterFormSubmit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterPublish Method

This hook is called after a publish is completed in the CMS.

C#

```
public virtual bool OnAfterPublish();
```

Visual Basic

```
Public virtual Function OnAfterPublish() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterReviewRejected Method

This hook is called after a review is rejected.

C#

```
public virtual bool OnAfterReviewRejected();
```

Visual Basic

```
Public virtual Function OnAfterReviewRejected() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterUserAdd Method

This hook is called after a user is created in the CMS.

C#

```
public virtual bool OnAfterUserAdd();
```

Visual Basic

```
Public virtual Function OnAfterUserAdd() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterUserDelete Method

This hook is called after a user is deleted from the CMS.

C#

```
public virtual bool OnAfterUserDelete();
```

Visual Basic

```
Public virtual Function OnAfterUserDelete() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterUserEdit Method

This hook is called after a user is edited in the CMS.

C#

```
public virtual bool OnAfterUserEdit();
```

Visual Basic

```
Public virtual Function OnAfterUserEdit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterUserLogin Method

This hook is called after a user logs into in the CMS.

C#

```
public virtual bool OnAfterUserLogin();
```

Visual Basic

```
Public virtual Function OnAfterUserLogin() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeActionRequestSent Method

This hook is called before an ActionRequest is Sent.

C#

```
public virtual bool OnBeforeActionRequestSent();
```

Visual Basic

```
Public virtual Function OnBeforeActionRequestSent() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeContentAdd Method

This hook is called before content is added to the CMS.

C#

```
public virtual bool OnBeforeContentAdd();
```

Visual Basic

```
Public virtual Function OnBeforeContentAdd() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeContentDelete Method

This hook is called before a delete of content in the CMS.

C#

```
public virtual bool OnBeforeContentDelete();
```

Visual Basic

```
Public virtual Function OnBeforeContentDelete() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeContentEdit Method

This hook is called before a content is edited in the CMS.

C#

```
public virtual bool OnBeforeContentEdit();
```

Visual Basic

```
Public virtual Function OnBeforeContentEdit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeContentFlagged Method

This hook is called before an Content (see page 580) is flagged.

C#

```
public virtual bool OnBeforeContentFlagged();
```

Visual Basic

```
Public virtual Function OnBeforeContentFlagged() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeFolderAdd Method

This hook is called before a folder is created in the CMS.

C#

```
public virtual bool OnBeforeFolderAdd();
```

Visual Basic

```
Public virtual Function OnBeforeFolderAdd() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeFolderDelete Method

This hook is called before a folder is deleted from the CMS.

C#

```
public virtual bool OnBeforeFolderDelete();
```

Visual Basic

```
Public virtual Function OnBeforeFolderDelete() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeFolderEdit Method

This hook is called before a folder is edited in the CMS.

C#

```
public virtual bool OnBeforeFolderEdit();
```

Visual Basic

```
Public virtual Function OnBeforeFolderEdit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeFormSubmit Method

This hook is called before form data is submitted to the CMS database.

C#

```
public virtual bool OnBeforeFormSubmit();
```

Visual Basic

```
Public virtual Function OnBeforeFormSubmit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforePublish Method

This hook is called before a publish takes place in the CMS.

C#

```
public virtual bool OnBeforePublish();
```

Visual Basic

```
Public virtual Function OnBeforePublish() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeReviewRejected Method

This hook is called before a review is rejected.

C#

```
public virtual bool OnBeforeReviewRejected();
```

Visual Basic

```
Public virtual Function OnBeforeReviewRejected() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeUserAdd Method

This hook is called before a user is created in the CMS.

C#

```
public virtual bool OnBeforeUserAdd();
```

Visual Basic

```
Public virtual Function OnBeforeUserAdd() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeUserDelete Method

This hook is called before a user is delete from the CMS.

C#

```
public virtual bool OnBeforeUserDelete();
```

Visual Basic

```
Public virtual Function OnBeforeUserDelete() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeUserEdit Method

This hook is called before a user is edited in the CMS.

C#

```
public virtual bool OnBeforeUserEdit();
```

Visual Basic

```
Public virtual Function OnBeforeUserEdit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeUserLogin Method

This hook is called before a user login is initiated in the CMS.

C#

```
public virtual bool OnBeforeUserLogin();
```

Visual Basic

```
Public virtual Function OnBeforeUserLogin() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent Properties

The properties of the ExtensionEvent class are listed here.

Protected Properties

	Name	Description
	Content (see page 580)	Returns object containing information about the Content, when the operation is content based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.
	Folder (see page 581)	Returns object containing information about the Folder, when the operation is folder based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.
	Form (see page 581)	Returns object containing information about the Form, when the operation is form based.
	Site (see page 581)	Returns object containing information about the Site that fired the event.
	User (see page 581)	Returns object containing information about the User, when the operation is user based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.

ExtensionEvent.Content Property

Returns object containing information about the Content, when the operation is content based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.

C#

```
protected ContentInfo Content;
```

Visual Basic

```
Protected Property Content() As ContentInfo
```

ExtensionEvent.Folder Property

Returns object containing information about the Folder, when the operation is folder based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.

C#

```
protected FolderInfo Folder;
```

Visual Basic

```
Protected Property Folder() As FolderInfo
```

ExtensionEvent.Form Property

Returns object containing information about the Form, when the operation is form based.

C#

```
protected FormInfo Form;
```

Visual Basic

```
Protected Property Form() As FormInfo
```

ExtensionEvent.Site Property

Returns object containing information about the Site that fired the event.

C#

```
protected SiteInfo Site;
```

Visual Basic

```
Protected Property Site() As SiteInfo
```

ExtensionEvent.User Property

Returns object containing information about the User, when the operation is user based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.

C#

```
protected UserInfo User;
```

Visual Basic

```
Protected Property User() As UserInfo
```

FlagInfo Class

Class which holds information about the flagged content. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

Class Hierarchy

Ektron.Cms.Extensibility.FlagInfo

C#

```
public sealed class FlagInfo;
```

Visual Basic

```
Public sealed Class FlagInfo
```

File

CmsInfoObjects.cs

FlagInfo Properties

The properties of the FlagInfo class are listed here.

Public Properties

	Name	Description
	ContentHtml (see page 582)	Gets a string containing all the html displayed to end users for a piece of content. (ReadOnly)
	ContentID (see page 582)	Gets a numeric value representing the content ID of the content. (ReadOnly)
	ContentLanguage (see page 583)	Gets an integer representing the language of the content. (ReadOnly)
	ContentTitle (see page 583)	Gets a string containing the title of the content. (ReadOnly)
	EntryId (see page 583)	Gets a numeric value representing the flagged entry ID. (ReadOnly)
	ErrorMessage (see page 583)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	FlagComment (see page 583)	Gets an string representing the flagged comment. (ReadOnly)
	FlagDate (see page 583)	Gets a datetime representing the flagged date time. (ReadOnly)
	FlagId (see page 584)	Gets a numeric value representing the flag ID associated to this content. (ReadOnly)
	FlagName (see page 584)	Gets an string representing the flag name associated to this content. (ReadOnly)
	UserAvatar (see page 584)	Gets an string representing path of the user's avatar. (ReadOnly)
	UserDisplayName (see page 584)	Gets an string representing the user's display name. (ReadOnly)
	UserFirstName (see page 584)	Gets an string representing the user's first name. (ReadOnly)
	UserID (see page 584)	Gets a numeric value representing the user ID of the user. (ReadOnly)
	UserLastName (see page 585)	Gets an string representing the user's last name. (ReadOnly)
	UserName (see page 585)	Gets an string representing the user's CMS user name. (ReadOnly)
	VisitorID (see page 585)	Gets an string representing the cookie value of the visitor. (ReadOnly)

FlagInfo.ContentHtml Property

Gets a string containing all the html displayed to end users for a piece of content. (ReadOnly)

C#

```
public string ContentHtml;
```

Visual Basic

```
Public Property ContentHtml() As string
```

FlagInfo.ContentID Property

Gets a numeric value representing the content ID of the content. (ReadOnly)

C#

```
public long ContentID;
```

Visual Basic

```
Public Property ContentID() As long
```

FlagInfo.ContentLanguage Property

Gets an integer representing the language of the content. (ReadOnly)

C#

```
public int ContentLanguage;
```

Visual Basic

```
Public Property ContentLanguage() As Integer
```

FlagInfo.ContentTitle Property

Gets a string containing the title of the content. (ReadOnly)

C#

```
public string ContentTitle;
```

Visual Basic

```
Public Property ContentTitle() As string
```

FlagInfo.EntryId Property

Gets a numeric value representing the flagged entry ID. (ReadOnly)

C#

```
public long EntryId;
```

Visual Basic

```
Public Property EntryId() As long
```

FlagInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

C#

```
public string ErrorMessage;
```

Visual Basic

```
Public Property ErrorMessage() As string
```

FlagInfo.FlagComment Property

Gets an string representing the flagged comment. (ReadOnly)

C#

```
public string FlagComment;
```

Visual Basic

```
Public Property FlagComment() As string
```

FlagInfo.FlagDate Property

Gets a datetime representing the flagged date time. (ReadOnly)

C#

```
public DateTime FlagDate;
```

Visual Basic

```
Public Property FlagDate() As DateTime
```

FlagInfo.FlagId Property

Gets a numeric value representing the flag ID associated to this content. (ReadOnly)

C#

```
public long FlagId;
```

Visual Basic

```
Public Property FlagId() As long
```

FlagInfo.FlagName Property

Gets an string representing the flag name associated to this content. (ReadOnly)

C#

```
public string FlagName;
```

Visual Basic

```
Public Property FlagName() As string
```

FlagInfo.UserAvatar Property

Gets an string representing path of the user's avatar. (ReadOnly)

C#

```
public string UserAvatar;
```

Visual Basic

```
Public Property UserAvatar() As string
```

FlagInfo.UserDisplayName Property

Gets an string representing the user's display name. (ReadOnly)

C#

```
public string UserDisplayName;
```

Visual Basic

```
Public Property UserDisplayName() As string
```

FlagInfo.UserFirstName Property

Gets an string representing the user's first name. (ReadOnly)

C#

```
public string UserFirstName;
```

Visual Basic

```
Public Property UserFirstName() As string
```

FlagInfo.UserID Property

Gets a numeric value representing the user ID of the user. (ReadOnly)

C#

```
public long UserID;
```

Visual Basic

```
Public Property UserID() As long
```

FlagInfo.UserLastName Property

Gets an string representing the user's last name. (ReadOnly)

C#

```
public string UserLastName;
```

Visual Basic

```
Public Property UserLastName() As string
```

FlagInfo.UserName Property

Gets an string representing the user's CMS user name. (ReadOnly)

C#

```
public string UserName;
```

Visual Basic

```
Public Property UserName() As string
```

FlagInfo.VisitorID Property

Gets an string representing the cookie value of the visitor. (ReadOnly)

C#

```
public string VisitorID;
```

Visual Basic

```
Public Property VisitorID() As string
```

FolderInfo Class

Class which holds information about the folder being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

Class Hierarchy

Ektron.Cms.Extensibility.FolderInfo

C#

```
public sealed class FolderInfo;
```

Visual Basic

```
Public sealed Class FolderInfo
```

File

CmsInfoObjects.cs

FolderInfo Properties

The properties of the FolderInfo class are listed here.

Public Properties

	Name	Description
	Description (see page 586)	Gets a string containing the description of the folder. (ReadOnly)
	ErrorMessage (see page 586)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	ID (see page 586)	Gets a numeric value representing the folder ID of the folder. (ReadOnly)

	Name (see page 586)	Gets a string containing the folder name. (ReadOnly)
	ParentID (see page 586)	Gets a numeric value representing the folder ID (see page 586) of the parent folder. (ReadOnly)
	ReplicationState (see page 586)	Gets an integer representing if replication is turned on for this folder. (ReadOnly)
	StyleSheet (see page 587)	Gets a string containing the style sheet to be applied to content in this folder. (ReadOnly)
	Type (see page 587)	Gets a string containing the folder type. (ReadOnly)

FolderInfo.Description Property

Gets a string containing the description of the folder. (ReadOnly)

C#

```
public string Description;
```

Visual Basic

```
Public Property Description() As string
```

FolderInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

C#

```
public string ErrorMessage;
```

Visual Basic

```
Public Property ErrorMessage() As string
```

FolderInfo.ID Property

Gets a numeric value representing the folder ID of the folder. (ReadOnly)

C#

```
public long ID;
```

Visual Basic

```
Public Property ID() As long
```

FolderInfo.Name Property

Gets a string containing the folder name. (ReadOnly)

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

FolderInfo.ParentID Property

Gets a numeric value representing the folder ID (see page 586) of the parent folder. (ReadOnly)

C#

```
public long ParentID;
```

Visual Basic

```
Public Property ParentID() As long
```

FolderInfo.ReplicationState Property

Gets an integer representing if replication is turned on for this folder. (ReadOnly)

C#

```
public int ReplicationState;
```

Visual Basic

```
Public Property ReplicationState() As Integer
```

FolderInfo.StyleSheet Property

Gets a string containing the style sheet to be applied to content in this folder. (ReadOnly)

C#

```
public string StyleSheet;
```

Visual Basic

```
Public Property StyleSheet() As string
```

FolderInfo.Type Property

Gets a string containing the folder type. (ReadOnly)

C#

```
public string Type;
```

Visual Basic

```
Public Property Type() As string
```

FormField Class

Class that describes a field in a form - name / value pair

Class Hierarchy

Ektron.Cms.Extensibility.FormField

C#

```
public sealed class FormField;
```

Visual Basic

```
Public sealed Class FormField
```

File

CmsInfoObjects.cs

FormField.FormField Constructor

C#

```
public FormField(string FieldName, string FieldValue);
```

Visual Basic

```
Public Sub New(FieldName As string, FieldValue As string)
```

Parameters

Parameters	Description
string FieldName	
string FieldValue	

FormField Properties

The properties of the FormField class are listed here.

Public Properties

	Name	Description
	FieldName (see page 588)	Gets the name of the field as a string
	FieldValue (see page 588)	Gets the value of the field as a string

FormField.FieldName Property

Gets the name of the field as a string

C#

```
public string FieldName;
```

Visual Basic

```
Public Property FieldName() As string
```

FormField.FieldValue Property

Gets the value of the field as a string

C#

```
public string FieldValue;
```

Visual Basic

```
Public Property FieldValue() As string
```

FormInfo Class

Class which describes a Form (see page 871) object

Class Hierarchy

Ektron.Cms.Extensibility.FormInfo

C#

```
public sealed class FormInfo;
```

Visual Basic

```
Public sealed Class FormInfo
```

File

CmsInfoObjects.cs

FormInfo Methods

The methods of the FormInfo class are listed here.

Public Methods

	Name	Description
	this (see page 588)	Gets a single FormField (see page 587) object indexed by Field Name

FormInfo.this Indexer

Gets a single FormField (see page 587) object indexed by Field Name

C#

```
public FormField this[string fieldName];
```

Visual Basic

```
Public Sub this(fieldName As string)
```

Parameters

Parameters	Description
string fieldName	Name of field

FormInfo Properties

The properties of the FormInfo class are listed here.

Public Properties

	Name	Description
	ErrorMessage (see page 589)	Gets or sets a string containing message to display to the end user if plug-in returns false. (ReadWrite)
	FormDescription (see page 589)	Gets the description of the form as a string
	FormFields (see page 589)	Gets a collection of the Form (see page 871) Field objects as a System.Collections.Generic.List
	FormTitle (see page 589)	Gets the title for the form as a string

FormInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if plug-in returns false. (ReadWrite)

C#

```
public string ErrorMessage;
```

Visual Basic

```
Public Property ErrorMessage() As string
```

FormInfo.FormDescription Property

Gets the description of the form as a string

C#

```
public string FormDescription;
```

Visual Basic

```
Public Property FormDescription() As string
```

FormInfo.FormFields Property

Gets a collection of the Form (see page 871) Field objects as a System.Collections.Generic.List

C#

```
public List<FormField> FormFields;
```

Visual Basic

```
Public Property FormFields() As List<FormField>
```

FormInfo.FormTitle Property

Gets the title for the form as a string

C#

```
public string FormTitle;
```

Visual Basic

```
Public Property FormTitle() As string
```

MessageBoardInfo Class

Class which holds information about the Invitation Recipient being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

Class Hierarchy

Ektron.Cms.Extensibility.MessageBoardInfo

C#

```
public sealed class MessageBoardInfo;
```

Visual Basic

```
Public sealed Class MessageBoardInfo
```

File

CmsInfoObjects.cs

MessageBoardInfo.MessageBoardInfo Constructor

Constructor

C#

```
public MessageBoardInfo(XmlNode xml);
```

Visual Basic

```
Public Sub New(xml As XmlNode)
```

Parameters

Parameters	Description
XmlNode xml	node containing Recipient info.

MessageBoardInfo Properties

The properties of the MessageBoardInfo class are listed here.

Public Properties

	Name	Description
	ErrorMessage (see page 590)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

MessageBoardInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

C#

```
public string ErrorMessage;
```

Visual Basic

```
Public Property ErrorMessage() As string
```

RequestInfo Class

Reserved for future use.

Class Hierarchy

Ektron.Cms.Extensibility.RequestInfo

C#

```
public sealed class RequestInfo;
```

Visual Basic

```
Public sealed Class RequestInfo
```

File

CmsInfoObjects.cs

ReviewInfo Class

Class which holds information about the review being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

Class Hierarchy

Ektron.Cms.Extensibility.ReviewInfo

C#

```
public sealed class ReviewInfo;
```

Visual Basic

```
Public sealed Class ReviewInfo
```

File

CmsInfoObjects.cs

ReviewInfo Properties

The properties of the ReviewInfo class are listed here.

Public Properties

	Name	Description
	AverageRating (see page 592)	Gets an decimal representing the average rating. (ReadOnly)
	ContentHtml (see page 592)	Gets a string containing all the html displayed to end users for a piece of content. (ReadOnly)
	ContentID (see page 592)	Gets a numeric value representing the content ID (see page 593) of the content. (ReadOnly)
	ContentLanguage (see page 592)	Gets an integer representing the language of the content. (ReadOnly)
	ContentTitle (see page 592)	Gets a string containing the title of the content. (ReadOnly)
	ErrorMessage (see page 593)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	ID (see page 593)	Gets a numeric value representing the review ID. (ReadOnly)
	NumberofRatings (see page 593)	Gets an integer representing the number of ratings. (ReadOnly)
	Rating (see page 593)	Gets an integer representing the rating. (ReadOnly)
	RatingDate (see page 593)	Gets a datetime representing the rating time. (ReadOnly)
	State (see page 593)	Gets an integer representing the state. (ReadOnly)

	UserAvatar (see page 594)	Gets an string representing path of the user's avatar. (ReadOnly)
	UserComments (see page 594)	Gets an string representing the review text. (ReadOnly)
	UserDisplayName (see page 594)	Gets an string representing the user's display name. (ReadOnly)
	UserFirstName (see page 594)	Gets an string representing the user's first name. (ReadOnly)
	UserID (see page 594)	Gets a numeric value representing the user ID (see page 593) of the user. (ReadOnly)
	UserLastName (see page 594)	Gets an string representing the user's last name. (ReadOnly)
	UserName (see page 595)	Gets an string representing the user's CMS user name. (ReadOnly)
	VisitorID (see page 595)	Gets an string representing the cookie value of the visitor. (ReadOnly)

ReviewInfo.AverageRating Property

Gets an decimal representing the average rating. (ReadOnly)

C#

```
public decimal AverageRating;
```

Visual Basic

```
Public Property AverageRating() As decimal
```

ReviewInfo.ContentHtml Property

Gets a string containing all the html displayed to end users for a piece of content. (ReadOnly)

C#

```
public string ContentHtml;
```

Visual Basic

```
Public Property ContentHtml() As string
```

ReviewInfo.ContentID Property

Gets a numeric value representing the content ID (see page 593) of the content. (ReadOnly)

C#

```
public long ContentID;
```

Visual Basic

```
Public Property ContentID() As long
```

ReviewInfo.ContentLanguage Property

Gets an integer representing the language of the content. (ReadOnly)

C#

```
public int ContentLanguage;
```

Visual Basic

```
Public Property ContentLanguage() As Integer
```

ReviewInfo.ContentTitle Property

Gets a string containing the title of the content. (ReadOnly)

C#

```
public string ContentTitle;
```

Visual Basic

```
Public Property ContentTitle() As string
```

ReviewInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

C#

```
public string ErrorMessage;
```

Visual Basic

```
Public Property ErrorMessage() As string
```

ReviewInfo.ID Property

Gets a numeric value representing the review ID. (ReadOnly)

C#

```
public long ID;
```

Visual Basic

```
Public Property ID() As long
```

ReviewInfo.NumberofRatings Property

Gets an integer representing the number of ratings. (ReadOnly)

C#

```
public int NumberofRatings;
```

Visual Basic

```
Public Property NumberofRatings() As Integer
```

ReviewInfo.Rating Property

Gets an integer representing the rating. (ReadOnly)

C#

```
public int Rating;
```

Visual Basic

```
Public Property Rating() As Integer
```

ReviewInfo.RatingDate Property

Gets a datetime representing the rating time. (ReadOnly)

C#

```
public DateTime RatingDate;
```

Visual Basic

```
Public Property RatingDate() As DateTime
```

ReviewInfo.State Property

Gets an integer representing the state. (ReadOnly)

C#

```
public int State;
```

Visual Basic

```
Public Property State() As Integer
```

ReviewInfo.UserAvatar Property

Gets an string representing path of the user's avatar. (ReadOnly)

C#

```
public string UserAvatar;
```

Visual Basic

```
Public Property UserAvatar() As string
```

ReviewInfo.UserComments Property

Gets an string representing the review text. (ReadOnly)

C#

```
public string UserComments;
```

Visual Basic

```
Public Property UserComments() As string
```

ReviewInfo.UserDisplayName Property

Gets an string representing the user's display name. (ReadOnly)

C#

```
public string UserDisplayName;
```

Visual Basic

```
Public Property UserDisplayName() As string
```

ReviewInfo.UserFirstName Property

Gets an string representing the user's first name. (ReadOnly)

C#

```
public string UserFirstName;
```

Visual Basic

```
Public Property UserFirstName() As string
```

ReviewInfo.UserID Property

Gets a numeric value representing the user ID (see page 593) of the user. (ReadOnly)

C#

```
public long UserID;
```

Visual Basic

```
Public Property UserID() As long
```

ReviewInfo.UserLastName Property

Gets an string representing the user's last name. (ReadOnly)

C#

```
public string UserLastName;
```

Visual Basic

```
Public Property UserLastName() As string
```

ReviewInfo.UserName Property

Gets an string representing the user's CMS user name. (ReadOnly)

C#

```
public string UserName;
```

Visual Basic

```
Public Property UserName() As string
```

ReviewInfo.VisitorID Property

Gets an string representing the cookie value of the visitor. (ReadOnly)

C#

```
public string VisitorID;
```

Visual Basic

```
Public Property VisitorID() As string
```

SiteInfo Class

Class which holds information about the site in which the event was fired.

Class Hierarchy

Ektron.Cms.Extensibility.SiteInfo

C#

```
public sealed class SiteInfo;
```

Visual Basic

```
Public sealed Class SiteInfo
```

File

CmsInfoObjects.cs

SiteInfo Properties

The properties of the SiteInfo class are listed here.

Public Properties

	Name	Description
	AppPath (see page 595)	Gets a string containing the full application path of asp.net application (ReadOnly)
	IsSSL (see page 596)	Gets a bool indicating if calling site uses SSL. (ReadOnly)
	Name (see page 596)	Gets a string containing the name of the calling site. (ReadOnly)
	PhysicalSitePath (see page 596)	Gets a string containing the physical path of the application (ReadOnly)
	Port (see page 596)	Gets an integer representing the port the calling site is running on. (ReadOnly)
	Url (see page 596)	Gets a string containing the URL of the calling site. (ReadOnly)

SiteInfo.AppPath Property

Gets a string containing the full application path of asp.net application (ReadOnly)

C#

```
public string AppPath;
```

Visual Basic

```
Public Property AppPath() As string
```

SiteInfo.IsSSL Property

Gets a bool indicating if calling site uses SSL. (ReadOnly)

C#

```
public bool IsSSL;
```

Visual Basic

```
Public Property IsSSL() As bool
```

SiteInfo.Name Property

Gets a string containing the name of the calling site. (ReadOnly)

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

SiteInfo.PhysicalSitePath Property

Gets a string containing the physical path of the application (ReadOnly)

C#

```
public string PhysicalSitePath;
```

Visual Basic

```
Public Property PhysicalSitePath() As string
```

SiteInfo.Port Property

Gets an integer representing the port the calling site is running on. (ReadOnly)

C#

```
public int Port;
```

Visual Basic

```
Public Property Port() As Integer
```

SiteInfo.Url Property

Gets a string containing the URL of the calling site. (ReadOnly)

C#

```
public string Url;
```

Visual Basic

```
Public Property Url() As string
```

UserInfo Class

Class which holds information about the user being processed. Certain hooks do not expose all information from the CMS,

default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

Class Hierarchy

Ektron.Cms.Extensibility.UserInfo

C#

```
public sealed class UserInfo;
```

Visual Basic

```
Public sealed Class UserInfo
```

File

CmsInfoObjects.cs

UserInfo Properties

The properties of the UserInfo class are listed here.

Public Properties

	Name	Description
	Addres (see page 597)	Gets a string containing the user addres
	DisplayName (see page 597)	Gets a string containing the display name of the user being added, edited or deleted. (ReadOnly)
	Editor (see page 598)	Gets a string containing the editor preference of the user being added, edited or deleted. (ReadOnly)
	EmailAddress (see page 598)	Gets a string containing the email address of the user being added, edited or deleted. (ReadOnly)
	ErrorMessage (see page 598)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	FirstName (see page 598)	Gets a string containing the first name of the user being added, edited or deleted. (ReadOnly)
	ID (see page 598)	Gets a numeric value representing the user ID of the being added, edited or deleted. (ReadOnly)
	Language (see page 598)	Gets a string containing the language preference of the user being added, edited or deleted. (ReadOnly)
	LastName (see page 599)	Gets a string containing the last name of the user being added, edited or deleted. (ReadOnly)
	Latitude (see page 599)	Gets a double value of map co-ordinate latitude of user address
	Longitude (see page 599)	Gets a double value of map co-ordinate longitude of user address
	Notifications (see page 599)	Gets an integer representing if the user has turned on email notifications of the user being added, edited or deleted. (ReadOnly)
	Type (see page 599)	Integer representing the user type of the user being added, edited or deleted. Only valid with OnBeforeUserAdd and OnAfterUserAdd. (ReadOnly)
	UserName (see page 599)	Gets a string containing the login name of the user being added, edited or deleted. (ReadOnly)

UserInfo.Addres Property

Gets a string containing the user addres

C#

```
public string Addres;
```

Visual Basic

```
Public Property Addres() As string
```

UserInfo.DisplayName Property

Gets a string containing the display name of the user being added, edited or deleted. (ReadOnly)

C#

```
public string DisplayName;
```

Visual Basic

```
Public Property DisplayName() As string
```

UserInfo.Editor Property

Gets a string containing the editor preference of the user being added, edited or deleted. (ReadOnly)

C#

```
public string Editor;
```

Visual Basic

```
Public Property Editor() As string
```

UserInfo.EmailAddress Property

Gets a string containing the email address of the user being added, edited or deleted. (ReadOnly)

C#

```
public string EmailAddress;
```

Visual Basic

```
Public Property EmailAddress() As string
```

UserInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

C#

```
public string ErrorMessage;
```

Visual Basic

```
Public Property ErrorMessage() As string
```

UserInfo.FirstName Property

Gets a string containing the first name of the user being added, edited or deleted. (ReadOnly)

C#

```
public string FirstName;
```

Visual Basic

```
Public Property FirstName() As string
```

UserInfo.ID Property

Gets a numeric value representing the user ID of the being added, edited or deleted. (ReadOnly)

C#

```
public long ID;
```

Visual Basic

```
Public Property ID() As long
```

UserInfo.Language Property

Gets a string containing the language preference of the user being added, edited or deleted. (ReadOnly)

C#

```
public string Language;
```

Visual Basic

```
Public Property Language() As string
```

UserInfo.LastName Property

Gets a string containing the last name of the user being added, edited or deleted. (ReadOnly)

C#

```
public string LastName;
```

Visual Basic

```
Public Property LastName() As string
```

UserInfo.Latitude Property

Gets a double value of map co-ordinate latitude of user address

C#

```
public double Latitude;
```

Visual Basic

```
Public Property Latitude() As double
```

UserInfo.Longitude Property

Gets a double value of map co-ordinate longitude of user address

C#

```
public double Longitude;
```

Visual Basic

```
Public Property Longitude() As double
```

UserInfo.Notifications Property

Gets an integer representing if the user has turned on email notifications of the user being added, edited or deleted. (ReadOnly)

C#

```
public int Notifications;
```

Visual Basic

```
Public Property Notifications() As Integer
```

UserInfo.Type Property

Integer representing the user type of the user being added, edited or deleted. Only valid with OnBeforeUserAdd and OnAfterUserAdd. (ReadOnly)

C#

```
public int Type;
```

Visual Basic

```
Public Property Type() As Integer
```

UserInfo.UserName Property

Gets a string containing the login name of the user being added, edited or deleted. (ReadOnly)

C#

```
public string UserName;
```

Visual Basic

```
Public Property UserName() As string
```

.NET Assemblies

The .NET Assemblies are used to access functionality typically provided in the Workarea. For example, adding content, deleting a user or retrieving information about a specific folder. By using the .NET Assemblies you can create custom functionality while keeping within the Ektron Inc. Business Rules.

This section is divided into separate subsections. The first subsection in this section contains classes derived from the Ektron.Cms.API (see page 669) Namespace.

- Ektron.Cms.API.BusinessRule (see page 671)
- Ektron.Cms.API.CustomFields (see page 673)
- Ektron.Cms.API.DateTimeSelector
- Ektron.Cms.API.Folder (see page 683)
- Ektron.Cms.API.Font
- Ektron.Cms.API.Library (see page 695)
- Ektron.Cms.API.Localization (see page 704)
- Ektron.Cms.API.Metadata (see page 714)
- Ektron.Cms.API.Permissions (see page 721)
- Ektron.Cms.API.Site (see page 725)
- Ektron.Cms.API.SiteMap (see page 732)
- Ektron.Cms.API.Subscription (see page 736)
- Ektron.Cms.API.Utilities (see page 744)

The rest of the subsections contain different Namespaces with classes below them.

- Ektron.Cms.API.Analytics.Analytics (see page 745)
- Ektron.Cms.API.Calendar.Calendar
- Ektron.Cms.API.Calendar.CalendarEvent
- Ektron.Cms.API.Calendar.CalendarEventType
- Ektron.Cms.API.Content.Asset (see page 822)
- Ektron.Cms.API.Content.Blog (see page 826)
- Ektron.Cms.API.Content.Content (see page 840)
- Ektron.Cms.API.Content.ContentRating (see page 866)
- Ektron.Cms.API.Content.Form (see page 871)
- Ektron.Cms.API.Content.Taxonomy (see page 876)
- Ektron.Cms.API.Content.ThreadedDiscussion (see page 882)
- Ektron.Cms.API.Search.SearchManager (see page 899)
- Ektron.Cms.API.Task.Task (see page 909)
- Ektron.Cms.API.Task.TaskCategory (see page 919)
- Ektron.Cms.API.Task.TaskCategoryType (see page 922)

- Ektron.Cms.API.User.ADUser (see page 951)
- Ektron.Cms.API.User.User (see page 958)

Namespaces

Name	Description
API (see page 669)	The API Namespace contains classes for programmatically accessing features typically found in the Workarea.
Framework (see page 995)	This is namespace Ektron.Cms.Framework.

eCommerce

This section includes API classes for eCommerce. These classes do not use the internal administrator to access their functionality. You must login a user or an administrator when using these methods. This is different from the rest of the API's contained in this documentation.

Classes

	Name	Description
	AddressApi (see page 602)	Api class for managing Addresses in Cms (see page 7).
	BasketApi (see page 605)	The API (see page 669) class used for managing Commerce (see page 1423) shopping carts (baskets).
	CatalogEntryApi (see page 611)	API (see page 669) class used for managing catalogs entries for ecommerce.
	ContractApi (see page 616)	Api class for managing Commerce (see page 1423) Contracts in Cms (see page 7).
	CountryApi (see page 618)	API (see page 669) class for managing countries in CMS400.NET.
	CouponApi (see page 620)	Api class for managing Commerce (see page 1423) Coupons in Cms (see page 7).
	CurrencyApi (see page 629)	Api class for managing commerce currencies in Cms (see page 7).
	CustomerApi (see page 631)	Api class for managing Commerce (see page 1423) Customers in Cms (see page 7).
	ExchangeRateApi (see page 634)	Api class for managing commerce exchange rates in Cms (see page 7).
	InventoryApi (see page 637)	Api class for managing Commerce (see page 1423) Inventory (see page 1407) in Cms (see page 7).
	OrderApi (see page 638)	Api class for managing Commerce (see page 1423) Orders in Cms (see page 7).
	PackageApi (see page 647)	API (see page 669) class used for managing shipping packages.
	ProductTypeApi (see page 649)	Api class for managing Commerce (see page 1423) product types in Cms (see page 7).
	RecommendationApi (see page 653)	Api class for managing entry reccomendations.
	RegionApi (see page 655)	Api class for managing regions in Cms (see page 7).
	ShippingMethodApi (see page 657)	Api class for managing shipping methods in CMS.
	TaxApi (see page 658)	Api class for managing Commerce (see page 1423) Tax rates in Cms (see page 7).
	TaxClassApi (see page 663)	Api class for managing Commerce (see page 1423) Tax Classes in Cms (see page 7).
	TaxTypeApi (see page 665)	Api class for managing Commerce (see page 1423) tax types in Cms (see page 7).
	WarehouseApi (see page 667)	This is class Ektron.Cms.Commerce.WarehouseApi.

Classes

The following table lists %CATEGORYL% in this documentation.

Description

This section includes API classes for eCommerce (see page 601). These classes do not use the internal administrator to access their functionality. You must login a user or an administrator when using these methods. This is different from the rest of the API's contained in this documentation.

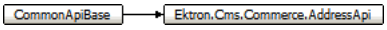
Classes

	Name	Description
	AddressApi (see page 602)	Api class for managing Addresses in Cms (see page 7).
	BasketApi (see page 605)	The API (see page 669) class used for managing Commerce (see page 1423) shopping carts (baskets).
	CatalogEntryApi (see page 611)	API (see page 669) class used for managing catalogs entries for ecommerce.
	ContractApi (see page 616)	Api class for managing Commerce (see page 1423) Contracts in Cms (see page 7).
	CountryApi (see page 618)	API (see page 669) class for managing countries in CMS400.NET.
	CouponApi (see page 620)	Api class for managing Commerce (see page 1423) Coupons in Cms (see page 7).
	CurrencyApi (see page 629)	Api class for managing commerce currencies in Cms (see page 7).
	CustomerApi (see page 631)	Api class for managing Commerce (see page 1423) Customers in Cms (see page 7).
	ExchangeRateApi (see page 634)	Api class for managing commerce exchange rates in Cms (see page 7).
	InventoryApi (see page 637)	Api class for managing Commerce (see page 1423) Inventory (see page 1407) in Cms (see page 7).
	OrderApi (see page 638)	Api class for managing Commerce (see page 1423) Orders in Cms (see page 7).
	PackageApi (see page 647)	API (see page 669) class used for managing shipping packages.
	ProductTypeApi (see page 649)	Api class for managing Commerce (see page 1423) product types in Cms (see page 7).
	RecommendationApi (see page 653)	Api class for managing entry reccomendations.
	RegionApi (see page 655)	Api class for managing regions in Cms (see page 7).
	ShippingMethodApi (see page 657)	Api class for managing shipping methods in CMS.
	TaxApi (see page 658)	Api class for managing Commerce (see page 1423) Tax rates in Cms (see page 7).
	TaxClassApi (see page 663)	Api class for managing Commerce (see page 1423) Tax Classes in Cms (see page 7).
	TaxTypeApi (see page 665)	Api class for managing Commerce (see page 1423) tax types in Cms (see page 7).
	WarehouseApi (see page 667)	This is class Ektron.Cms.Commerce.WarehouseApi.

AddressApi Class

Api class for managing Addresses in Cms (see page 7).

Class Hierarchy



C#

```
public class AddressApi : CommonApiBase;
```

Visual Basic

```
Public Class AddressApi
Inherits CommonApiBase
```

File

AddressApi.cs

AddressApi Methods

The methods of the AddressApi class are listed here.

Public Methods

	Name	Description
	Add (see page 603)	This is the overview for the Add method overload.
	Delete (see page 604)	Deletes an address.
	GetItem (see page 604)	Retrieves an AddressData (see page 1428) object by providing its ID.

	GetList (see page 604)	Retrieves a list of addresses for a given customer's ID.
	Update (see page 604)	Updates the supplied Address.
	UpdateOrderAddress (see page 605)	Updates an address associated with an order. This update only affects this specific address and does not affect the customer's actual saved address.

Add Method

This is the overview for the Add method overload.

Description

The GetDefaultBasket method is overloaded.

- Add Method (AddressData) (see page 603) - Adds a new address based on information in an AddressData object.
- Add Method (AddressData, long) (see page 603) - Adds a new address based on information in an AddressData object and associates with a customer.

Overload List

	Name	Description
	AddressApi.Add (AddressData) (see page 603)	Adds a new address based on information in an AddressData (see page 1428) object. AddressData.Id will be populated with the newly created ID.
	AddressApi.Add (AddressData, long) (see page 603)	Adds a new address based on information in an AddressData (see page 1428) object and associates with a customer. AddressData.Id will be populated with the newly created ID.

AddressApi.Add Method (AddressData)

Adds a new address based on information in an AddressData (see page 1428) object. AddressData.Id will be populated with the newly created ID.

C#

```
public long Add(AddressData addressData);
```

Visual Basic

```
Public Function Add(addressData As AddressData) As long
```

Parameters

Parameters	Description
AddressData addressData	The AddressData (see page 1428) object to add.

AddressApi.Add Method (AddressData, long)

Adds a new address based on information in an AddressData (see page 1428) object and associates with a customer. AddressData.Id will be populated with the newly created ID.

C#

```
public long Add(AddressData addressData, long customerId);
```

Visual Basic

```
Public Function Add(addressData As AddressData, customerId As long) As long
```

Parameters

Parameters	Description
AddressData addressData	The AddressData (see page 1428) object to add.
long customerId	The ID of customer to which this address belongs.

Remarks

Required: Name, AddressLine1, City, PostalCode, Phone.

AddressApi.Delete Method

Deletes an address.

C#

```
public void Delete(long addressId);
```

Visual Basic

```
Public Function Delete(addressId As long) As void
```

Parameters

Parameters	Description
long addressId	The ID of the AddressData (see page 1428) object to delete.

AddressApi.GetItem Method

Retrieves an AddressData (see page 1428) object by providing its ID.

C#

```
public AddressData GetItem(long addressId);
```

Visual Basic

```
Public Function GetItem(addressId As long) As AddressData
```

Parameters

Parameters	Description
long addressId	The ID the of AddressData (see page 1428) object to retrieve.

Returns

An AddressData (see page 1428) object.

AddressApi.GetList Method

Retrieves a list of addresses for a given customer's ID.

C#

```
public List<AddressData> GetList(long customerId);
```

Visual Basic

```
Public Function GetList(customerId As long) As List<AddressData>
```

Parameters

Parameters	Description
long customerId	The ID of the customer for which to retrieve a list of addresses.

Returns

An AddressData (see page 1428) object.

AddressApi.Update Method

Updates the supplied Address.

C#

```
public void Update(AddressData addressData);
```

Visual Basic

```
Public Function Update(addressData As AddressData) As void
```

Parameters

Parameters	Description
AddressData addressData	The ID of the AddressData (see page 1428) object to update.

AddressApi.UpdateOrderAddress Method

Updates an address associated with an order. This update only affects this specific address and does not affect the customer's actual saved address.

C#

```
public void UpdateOrderAddress(AddressData addressData, long orderId, bool isShipping, bool
isBilling);
```

Visual Basic

```
Public Function UpdateOrderAddress(addressData As AddressData, orderId As long, isShipping As
bool, isBilling As bool) As void
```

Parameters

Parameters	Description
AddressData addressData	Address info to update.
long orderId	Id of order to update address for.
bool isShipping	If true, updates the shipping address for an order.
bool isBilling	If true, updates the billing address for an order.

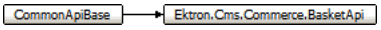
Description

///

BasketApi Class

The API (see page 669) class used for managing Commerce (see page 1423) shopping carts (baskets).

Class Hierarchy



C#

```
public class BasketApi : CommonApiBase;
```

Visual Basic

```
Public Class BasketApi
Inherits CommonApiBase
```

File

BasketApi.cs

BasketApi Methods

The methods of the BasketApi class are listed here.

Public Methods

	Name	Description
	CalculateBasket (see page 606)	This is the overview for the CalculateBasket method overload.
	CreateNewBasket (see page 607)	Creates a new cart for the supplied user ID. By setting the `openBasketYou` parameter to true, the new cart becomes the default open cart.
	Delete (see page 608)	Deletes a cart based on its ID.
	Empty (see page 608)	Deletes all the items in a cart based on the cart's ID.
	GetDefaultBasket (see page 608)	This is the overview for the GetDefaultBasket method overload.
	GetItem (see page 609)	Retrieves a single cart based on its ID.
	GetList (see page 610)	Retrieves a list saved carts for a user ID.
	SetDefaultBasket (see page 610)	This is the overview for the SetDefaultBasket method overload.

CalculateBasket Method

This is the overview for the CalculateBasket method overload.

Overload List

	Name	Description
	BasketApi.CalculateBasket (Basket) (see page 606)	Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc.
	BasketApi.CalculateBasket (Basket, AddressData) (see page 606)	Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc.
	BasketApi.CalculateBasket (long) (see page 607)	Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc.
	BasketApi.CalculateBasket (long, AddressData) (see page 607)	Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc.

BasketApi.CalculateBasket Method (Basket)

Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc.

C#

```
public BasketCalculator CalculateBasket(Basket basket);
```

Visual Basic

```
Public Function CalculateBasket(basket As Basket) As BasketCalculator
```

Parameters

Parameters	Description
Basket basket	Cart (see page 81) to calculate.

BasketApi.CalculateBasket Method (Basket, AddressData)

Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc.

C#

```
public BasketCalculator CalculateBasket(Basket basket, AddressData shippingAddress);
```

Visual Basic

```
Public Function CalculateBasket(basket As Basket, shippingAddress As AddressData) As
BasketCalculator
```

Parameters

Parameters	Description
Basket basket	Cart (see page 81) to calculate.
AddressData shippingAddress	Shipping Address to use to calculate taxes and shipping. This Address is not saved with the basket and is only used for the returned calculations.

BasketApi.CalculateBasket Method (long)

Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc.

C#

```
public BasketCalculator CalculateBasket(long basketId);
```

Visual Basic

```
Public Function CalculateBasket(basketId As long) As BasketCalculator
```

Parameters

Parameters	Description
long basketId	ID of the cart to calculate.

BasketApi.CalculateBasket Method (long, AddressData)

Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc.

C#

```
public BasketCalculator CalculateBasket(long basketId, AddressData shippingAddress);
```

Visual Basic

```
Public Function CalculateBasket(basketId As long, shippingAddress As AddressData) As
BasketCalculator
```

Parameters

Parameters	Description
long basketId	ID of the cart to calculate.
AddressData shippingAddress	Shipping Address to use to calculate taxes and shipping. This Address is not saved with the basket and is only used for the returned calculations.

BasketApi.CreateNewBasket Method

Creates a new cart for the supplied user ID. By setting the `openBasketYou` parameter to true, the new cart becomes the default open cart.

C#

```
public Basket CreateNewBasket(long userId, bool openBasket);
```

Visual Basic

```
Public Function CreateNewBasket(userId As long, openBasket As bool) As Basket
```

Parameters

Parameters	Description
long userId	ID of the user for which to create a cart.

bool openBasket	When set to true, the newly created cart becomes the default open cart.

Returns

Newly created cart object.

BasketApi.Delete Method

Deletes a cart based on its ID.

C#

```
public void Delete(long basketId);
```

Visual Basic

```
Public Function Delete(basketId As long) As void
```

Parameters

Parameters	Description
long basketId	The ID of cart to delete.

BasketApi.Empty Method

Deletes all the items in a cart based on the cart's ID.

C#

```
public void Empty(long basketId);
```

Visual Basic

```
Public Function Empty(basketId As long) As void
```

Parameters

Parameters	Description
long basketId	The ID of basket to empty.

GetDefaultBasket Method

This is the overview for the GetDefaultBasket method overload.

Description

The GetDefaultBasket method is overloaded.

- GetDefaultBasket Method () (see page 609) - Gets the default cart.
- GetDefaultBasket Method (long) (see page 609) - Gets the default cart based on a user's ID.
- GetDefaultBasket Method (string) (see page 609) - Gets the default cart based on a non logged in user's GUID shopping ID.

Overload List

	Name	Description
	BasketApi.GetDefaultBasket () (see page 609)	Gets the default cart.
	BasketApi.GetDefaultBasket (long) (see page 609)	Gets a default cart for a specified user.
	BasketApi.GetDefaultBasket (string) (see page 609)	Gets a default cart for a non logged in user based upon the shopper's GUID string.

BasketApi.GetDefaultBasket Method ()

Gets the default cart.

C#

```
public Basket GetDefaultBasket();
```

Visual Basic

```
Public Function GetDefaultBasket() As Basket
```

Returns

The default cart object.

BasketApi.GetDefaultBasket Method (long)

Gets a default cart for a specified user.

C#

```
public Basket GetDefaultBasket(long userId);
```

Visual Basic

```
Public Function GetDefaultBasket(userId As long) As Basket
```

Parameters

Parameters	Description
long userId	The ID of user for which to retrieve a basket.

Returns

A default cart object for a specified.

BasketApi.GetDefaultBasket Method (string)

Gets a default cart for a non logged in user based upon the shopper's GUID string.

C#

```
public Basket GetDefaultBasket(string shopperId);
```

Visual Basic

```
Public Function GetDefaultBasket(shopperId As string) As Basket
```

Parameters

Parameters	Description
string shopperId	The GUID ID for the non logged in shopper.

Returns

a default cart object for a non logged in user.

BasketApi.GetItem Method

Retrieves a single cart based on its ID.

C#

```
public Basket GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As Basket
```

Parameters

Parameters	Description
long id	ID of cart to retrieve.

Returns

A cart object.

BasketApi.GetList Method

Retrieves a list saved carts for a user ID.

C#

```
public List<Basket> GetList(long userId);
```

Visual Basic

```
Public Function GetList(userId As long) As List<Basket>
```

Parameters

Parameters	Description
long userId	ID of user for which to retrieve a list of saved carts.

Returns

A list of saved cart objects.

SetDefaultBasket Method

This is the overview for the SetDefaultBasket method overload.

Description

The SetDefaultBasket method is overloaded.

- SetDefaultBasket Method (long) (see page 610) - Sets the logged in user's default cart based on the cart's ID.
- SetDefaultBasket Method (long, long) (see page 611) - Sets the default cart based on the cart's ID and user's ID.

Overload List

	Name	Description
	BasketApi.SetDefaultBasket (long) (see page 610)	Sets the default cart for the current user.
	BasketApi.SetDefaultBasket (long, long) (see page 611)	Sets the default cart for a user.

BasketApi.SetDefaultBasket Method (long)

Sets the default cart for the current user.

C#

```
public void SetDefaultBasket(long basketId);
```

Visual Basic

```
Public Function SetDefaultBasket(basketId As long) As void
```

Parameters

Parameters	Description
long basketId	The ID of the cart to set as the default.

BasketApi.SetDefaultBasket Method (long, long)

Sets the default cart for a user.

C#

```
public void SetDefaultBasket(long basketId, long userId);
```

Visual Basic

```
Public Function SetDefaultBasket(basketId As long, userId As long) As void
```

Parameters

Parameters	Description
long basketId	The ID of the cart to set as the default.
long userId	The ID of the user setting the default cart.

CatalogEntryApi Class

API (see page 669) class used for managing catalogs entries for ecommerce.

Class Hierarchy

C#

```
public class CatalogEntryApi : CommonApiBase;
```

Visual Basic

```
Public Class CatalogEntryApi
Inherits CommonApiBase
```

File

CatalogApi.cs

CatalogEntryApi Methods

The methods of the CatalogEntryApi class are listed here.

Public Methods

	Name	Description
	Add (see page 612)	Adds and publishes a new catalog entry based on an EntryData (see page 1487) object.
	AddAndCheckIn (see page 612)	Adds and checks-in a new catalog entry based on an EntryData (see page 1487) object.
	AddAndPublish (see page 612)	Adds and publishes a new catalog entry based on an EntryData (see page 1487) object.
	AddAndSave (see page 612)	Adds and saves a new catalog entry based on an EntryData (see page 1487) object. **Note:** The catalog entry is not checked in or published.
	Checkout (see page 613)	Checks out a catalog entry based on its ID.
	Delete (see page 613)	Deletes a catalog entry based on its ID.
	GetAttributeEntryList (see page 613)	Retrieves a list of entrydata objects based upon EntryAttributeCriteria.
	GetItem (see page 613)	This is the overview for the GetItem method overload.
	GetItemEdit (see page 614)	Gets the saved or checked in version of the catalog entry. If status is A, then gets the currently published catalog entry. Does NOT check out the catalog entry automatically.

	GetList (see page 615)	Retrieves a list of EntryData (see page 1487) objects based upon the supplied criteria.
	GetTaxonomyEntryList (see page 615)	Retrieves a list of EntryData (see page 1487) objects in the supplied taxonomy based upon the supplied criteria.
	Restore (see page 615)	Restores a previous version of a catalog entry based on its ID and the version ID to restore.
	UndoCheckOut (see page 615)	Undoes a check out for a catalog entry based on its ID. **Note:** Any changes made to saved version will be lost.
	Update (see page 616)	Saves an existing catalog entry.
	UpdateAndCheckIn (see page 616)	Saves and checks-in an existing catalog entry.
	UpdateAndPublish (see page 616)	Saves and submits an existing catalog entry for publish.

CatalogEntryApi.Add Method

Adds and publishes a new catalog entry based on an EntryData (see page 1487) object.

C#

```
public void Add(EntryData entry);
```

Visual Basic

```
Public Function Add(entry As EntryData) As void
```

Parameters

Parameters	Description
EntryData entry	The EntryData (see page 1487) object to add.

CatalogEntryApi.AddAndCheckIn Method

Adds and checks-in a new catalog entry based on an EntryData (see page 1487) object.

C#

```
public void AddAndCheckIn(EntryData entry);
```

Visual Basic

```
Public Function AddAndCheckIn(entry As EntryData) As void
```

Parameters

Parameters	Description
EntryData entry	The EntryData (see page 1487) object to add and checkin.

CatalogEntryApi.AddAndPublish Method

Adds and publishes a new catalog entry based on an EntryData (see page 1487) object.

C#

```
public void AddAndPublish(EntryData entry);
```

Visual Basic

```
Public Function AddAndPublish(entry As EntryData) As void
```

Parameters

Parameters	Description
EntryData entry	The EntryData (see page 1487) object to add and publish.

CatalogEntryApi.AddAndSave Method

Adds and saves a new catalog entry based on an EntryData (see page 1487) object.

Note: The catalog entry is not checked in or published.

C#

```
public void AddAndSave(EntryData entry);
```

Visual Basic

```
Public Function AddAndSave(entry As EntryData) As void
```

Parameters

Parameters	Description
EntryData entry	The EntryData (see page 1487) object to add and save.

CatalogEntryApi.Checkout Method

Checks out a catalog entry based on its ID.

C#

```
public void Checkout(long entryId);
```

Visual Basic

```
Public Function Checkout(entryId As long) As void
```

Parameters

Parameters	Description
long entryId	The ID of catalog entry to check out.

CatalogEntryApi.Delete Method

Deletes a catalog entry based on its ID.

C#

```
public void Delete(long entryId);
```

Visual Basic

```
Public Function Delete(entryId As long) As void
```

Parameters

Parameters	Description
long entryId	The ID of entry to delete.

CatalogEntryApi.GetAttributeEntryList Method

Retrieves a list of entrydata objects based upon EntryAttributeCriteria.

C#

```
public List<EntryData> GetAttributeEntryList(EntryAttributeCriteria criteria);
```

Visual Basic

```
Public Function GetAttributeEntryList(criteria As EntryAttributeCriteria) As List<EntryData>
```

Parameters

Parameters	Description
EntryAttributeCriteria criteria	EntryAttributeCriteria by which to retrieve entry list.

GetItem Method

This is the overview for the GetItem method overload.

Description

The GetItem method is overloaded.

- GetItem Method (long) (see page 614) - Retrieves a single catalog entry in the current language based on the entry's ID.
- GetItem Method (long, long) (see page 614) - Retrieves a single catalog entry in the supplied language based on the catalog entry's ID and language ID.

Overload List

	Name	Description
	CatalogEntryApi.GetItem (long) (see page 614)	Retrieves a single catalog entry in the current language based on the entry's ID.
	CatalogEntryApi.GetItem (long, long) (see page 614)	Retrieves a single catalog entry in the supplied language based on the catalog entry's ID and language ID.

CatalogEntryApi.GetItem Method (long)

Retrieves a single catalog entry in the current language based on the entry's ID.

C#

```
public EntryData GetItem(long entryId);
```

Visual Basic

```
Public Function GetItem(entryId As long) As EntryData
```

Parameters

Parameters	Description
long entryId	The ID of catalog entry to retrieve.

CatalogEntryApi.GetItem Method (long, long)

Retrieves a single catalog entry in the supplied language based on the catalog entry's ID and language ID.

C#

```
public EntryData GetItem(long entryId, long entryLanguage);
```

Visual Basic

```
Public Function GetItem(entryId As long, entryLanguage As long) As EntryData
```

Parameters

Parameters	Description
long entryId	The ID of the catalog entry to retrieve.
long entryLanguage	The ID of the language for which to retrieve a catalog entry.

CatalogEntryApi.GetItemEdit Method

Gets the saved or checked in version of the catalog entry. If status is A, then gets the currently published catalog entry. Does NOT check out the catalog entry automatically.

C#

```
public EntryData GetItemEdit(long entryId, long entryLanguage, bool checkOut);
```

Visual Basic

```
Public Function GetItemEdit(entryId As long, entryLanguage As long, checkOut As bool) As
EntryData
```

Parameters

Parameters	Description
long entryId	The ID of the catalog entry.

CatalogEntryApi.GetList Method

Retrieves a list of EntryData (see page 1487) objects based upon the supplied criteria.

C#

```
public List<EntryData> GetList(Criteria<EntryProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<EntryProperty>) As List<EntryData>
```

Parameters

Parameters	Description
Criteria<EntryProperty> criteria	The criteria for retrieving EntryData (see page 1487).

CatalogEntryApi.GetTaxonomyEntryList Method

Retrieves a list of EntryData (see page 1487) objects in the supplied taxonomy based upon the supplied criteria.

C#

```
public List<EntryData> GetTaxonomyEntryList(long taxonomyId, int depth,
Criteria<EntryProperty> criteria);
```

Visual Basic

```
Public Function GetTaxonomyEntryList(taxonomyId As long, depth As Integer, criteria As
Criteria<EntryProperty>) As List<EntryData>
```

Parameters

Parameters	Description
long taxonomyId	The ID of taxonomy for which to reterieve EntryData (see page 1487) objects.
Criteria<EntryProperty> criteria	The criteria for retrieving EntryData (see page 1487).

CatalogEntryApi.Restore Method

Restores a previous version of a catalog entry based on its ID and the version ID to restore.

C#

```
public void Restore(long id, long versionId);
```

Visual Basic

```
Public Function Restore(id As long, versionId As long) As void
```

Parameters

Parameters	Description
long id	The ID of the catalog entry to Restore.
long versionId	Version Id to restore.

CatalogEntryApi.UndoCheckOut Method

Undoes a check out for a catalog entry based on its ID. **Note:** Any changes made to saved version will be lost.

C#

```
public void UndoCheckOut(long entryId);
```

Visual Basic

```
Public Function UndoCheckOut(entryId As long) As void
```

Parameters

Parameters	Description
long entryId	The ID of the catalog entry being reverted.

CatalogEntryApi.Update Method

Saves an existing catalog entry.

C#

```
public void Update(EntryData entry);
```

Visual Basic

```
Public Function Update(entry As EntryData) As void
```

Parameters

Parameters	Description
EntryData entry	The EntryData (see page 1487) object to save.

CatalogEntryApi.UpdateAndCheckIn Method

Saves and checks-in an existing catalog entry.

C#

```
public void UpdateAndCheckIn(EntryData entry);
```

Visual Basic

```
Public Function UpdateAndCheckIn(entry As EntryData) As void
```

Parameters

Parameters	Description
EntryData entry	The EntryData (see page 1487) object to save and check-in.

CatalogEntryApi.UpdateAndPublish Method

Saves and submits an existing catalog entry for publish.

C#

```
public void UpdateAndPublish(EntryData entry);
```

Visual Basic

```
Public Function UpdateAndPublish(entry As EntryData) As void
```

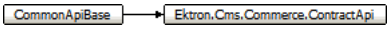
Parameters

Parameters	Description
EntryData entry	The EntryData (see page 1487) object to save and publish.

ContractApi Class

Api class for managing Commerce (see page 1423) Contracts in Cms (see page 7).

Class Hierarchy



C#

```
public class ContractApi : CommonApiBase;
```

Visual Basic

```
Public Class ContractApi
Inherits CommonApiBase
```

File

ContractApi.cs

ContractApi Methods

The methods of the ContractApi class are listed here.

Public Methods

	Name	Description
	Add (see page 617)	Saves a new ContractData (see page 1460) object. Returns the newly created Id.
	Delete (see page 617)	Deletes a ContractData (see page 1460) object.
	GetItem (see page 617)	Retrieves a ContractData (see page 1460) object by Id.
	GetList (see page 618)	Retrieves list of contract objects.
	Update (see page 618)	Updates the supplied ContractData (see page 1460) object.

ContractApi.Add Method

Saves a new ContractData (see page 1460) object. Returns the newly created Id.

C#

```
public void Add(ContractData contract);
```

Visual Basic

```
Public Function Add(contract As ContractData) As void
```

Parameters

Parameters	Description
ContractData contract	ContractData (see page 1460) to save.

ContractApi.Delete Method

Deletes a ContractData (see page 1460) object.

C#

```
public void Delete(long contractId);
```

Visual Basic

```
Public Function Delete(contractId As long) As void
```

Parameters

Parameters	Description
long contractId	Id of contract to delete

ContractApi.GetItem Method

Retrieves a ContractData (see page 1460) object by Id.

C#

```
public ContractData GetItem(long contractId);
```

Visual Basic

```
Public Function GetItem(contractId As long) As ContractData
```

Parameters

Parameters	Description
long contractId	Id of ContractData (see page 1460) object to retrieve.

ContractApi.GetList Method

Retrieves list of contract objects.

C#

```
public List<ContractData> GetList(Criteria<ContractProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<ContractProperty>) As List<ContractData>
```

Parameters

Parameters	Description
Criteria<ContractProperty> criteria	criteria for contract retrieval

ContractApi.Update Method

Updates the supplied ContractData (see page 1460) object.

C#

```
public void Update(ContractData contract);
```

Visual Basic

```
Public Function Update(contract As ContractData) As void
```

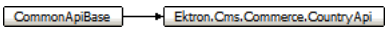
Parameters

Parameters	Description
ContractData contract	ContractData (see page 1460) object to update.

CountryApi Class

API (see page 669) class for managing countries in CMS400.NET.

Class Hierarchy



C#

```
public class CountryApi : CommonApiBase;
```

Visual Basic

```
Public Class CountryApi
Inherits CommonApiBase
```

File

CountryApi.cs

CountryApi Methods

The methods of the CountryApi class are listed here.

Public Methods

	Name	Description
	Add (see page 619)	Adds a new country to CMS400.NET based on the CountryData (see page 1461) object. The Country.Id property will be populated with the new Country's ID.
	CanDelete (see page 619)	Checks to see if a country can be deleted. A country cannot be deleted if a region or address uses the country.
	Delete (see page 619)	Deletes a country from the CMS400.NET based on its ID.
	GetItem (see page 620)	Retrieves a country by its ID.
	GetList (see page 620)	Retrieves a list of countries based on filters set in the `Criteria` property.
	Update (see page 620)	Updates an existing country in CMS400.NET based on the CountryData (see page 1461) object.

CountryApi.Add Method

Adds a new country to CMS400.NET based on the CountryData (see page 1461) object. The Country.Id property will be populated with the new Country's ID.

C#

```
public void Add(CountryData countryData);
```

Visual Basic

```
Public Function Add(countryData As CountryData) As void
```

Parameters

Parameters	Description
CountryData countryData	The CountryData (see page 1461) object to add.

CountryApi.CanDelete Method

Checks to see if a country can be deleted. A country cannot be deleted if a region or address uses the country.

C#

```
public bool CanDelete(int countryId, out ValidationResults result);
```

Visual Basic

```
Public Function CanDelete(countryId As Integer, result As out ValidationResults) As bool
```

Parameters

Parameters	Description
int countryId	The ID of the country to delete.
out ValidationResults result	ValidationResult returned containing messages if country cannot be deleted.

Returns

Returns true if country can safely be deleted.

CountryApi.Delete Method

Deletes a country from the CMS400.NET based on its ID.

C#

```
public void Delete(int id);
```

Visual Basic

```
Public Function Delete(id As Integer) As void
```

Parameters

Parameters	Description
int id	The ID of country to delete.

CountryApi.GetItem Method

Retrieves a country by its ID.

C#

```
public CountryData GetItem(int id);
```

Visual Basic

```
Public Function GetItem(id As Integer) As CountryData
```

Parameters

Parameters	Description
int id	The ID of country to retrieve.

Returns

A CountryData (see page 1461) object.

CountryApi.GetList Method

Retrieves a list of countries based on filters set in the `Criteria` property.

C#

```
public List<CountryData> GetList(Criteria<CountryProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<CountryProperty>) As List<CountryData>
```

Parameters

Parameters	Description
Criteria<CountryProperty> criteria	Criteria by which to filter the list of countries being retrieved.

Returns

A list of countries meeting criteria filters.

CountryApi.Update Method

Updates an existing country in CMS400.NET based on the CountryData (see page 1461) object.

C#

```
public void Update(CountryData countryData);
```

Visual Basic

```
Public Function Update(countryData As CountryData) As void
```

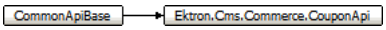
Parameters

Parameters	Description
CountryData countryData	The CountryData (see page 1461) object to save.

CouponApi Class

Api class for managing Commerce (see page 1423) Coupons in Cms (see page 7).

Class Hierarchy



C#

```
public class CouponApi : CommonApiBase;
```

Visual Basic

```
Public Class CouponApi
Inherits CommonApiBase
```

File

CouponApi.cs

CouponApi Methods

The methods of the CouponApi class are listed here.

Public Methods

	Name	Description
	Add (see page 621)	Saves an new couponData object. CouponData.Id will be populated with the newly created Id.
	AddCouponToCatalog (see page 622)	Applys a coupon to a catalog.
	AddCouponToProduct (see page 622)	Applys a coupon to a product.
	AddCouponToTaxonomy (see page 622)	Applys a coupon to a catalog.
	ApplyCouponToBasket (see page 623)	Applies a coupon to a customer's basket.
	CalculateBasketCoupons (see page 623)	This is the overview for the CalculateBasketCoupons method overload.
	Deactivate (see page 624)	Deactivate a coupondata object.
	Delete (see page 624)	Deletes a coupondata object.
	DeleteCouponProducts (see page 624)	Deletes all the products a coupon is applied to accept the products included in the ExclusionList.
	GetCatalogList (see page 624)	retrieves list of CouponEntryData (see page 1470) for catalogs with coupons applied to them.
	GetCouponsAppliedToBasket (see page 625)	Retrievies a list of coupons applied to a basket.
	GetItem (see page 625)	Retrieves a CouponData (see page 1466) object by Id.
	GetList (see page 625)	Retrieves list of couponData objects for a given customer.
	GetProductList (see page 625)	This is the overview for the GetProductList method overload.
	GetTaxonomyList (see page 626)	retrieves list of CouponEntryData (see page 1470) for taxonomies with coupons applied to them.
	IsCouponAppliedToProduct (see page 626)	Returns true if the coupon is applied to the catalog entry.
	IsCouponUsedForBasket (see page 627)	Checks if coupon has been applied to a basket.
	IsCouponUsedForOrder (see page 627)	Checks if coupon has been used by an order.
	SaveCouponApplications (see page 627)	Saves all the applications (taxonomy, catalog, product) for a given coupon. This is assumed to be an all inclusive lists. Any existing coupon applications that are not included will be deleted.
	Update (see page 628)	Updates the supplied CouponData (see page 1466) object.
	ValidateCoupon (see page 628)	This is the overview for the ValidateCoupon method overload.

CouponApi.Add Method

Saves an new couponData object. CouponData.Id will be populated with the newly created Id.

C#

```
public void Add(CouponData couponData);
```

Visual Basic

```
Public Function Add(couponData As CouponData) As void
```

Parameters

Parameters	Description
CouponData couponData	CouponData (see page 1466) to save.

CouponApi.AddCouponToCatalog Method

Applys a coupon to a catalog.

C#

```
public void AddCouponToCatalog(long couponId, long catalogId, bool includeCatalog);
```

Visual Basic

```
Public Function AddCouponToCatalog(couponId As long, catalogId As long, includeCatalog As
bool) As void
```

Parameters

Parameters	Description
long couponId	Id of coupon to apply.
long catalogId	Id of catalog to apply coupon to.
bool includeCatalog	if true, include catalog in coupon application. Else excludes catalog from coupon.

CouponApi.AddCouponToProduct Method

Applys a coupon to a product.

C#

```
public void AddCouponToProduct(long couponId, long productId, bool includeProduct);
```

Visual Basic

```
Public Function AddCouponToProduct(couponId As long, productId As long, includeProduct As
bool) As void
```

Parameters

Parameters	Description
long couponId	Id of coupon to apply.
long productId	Id of product to apply coupon to.
bool includeProduct	if true, include product in coupon application. Else excludes product from coupon.

CouponApi.AddCouponToTaxonomy Method

Applys a coupon to a catalog.

C#

```
public void AddCouponToTaxonomy(long couponId, long taxonomyId, bool includeTaxonomy);
```

Visual Basic

```
Public Function AddCouponToTaxonomy(couponId As long, taxonomyId As long, includeTaxonomy As
bool) As void
```

Parameters

Parameters	Description
long couponId	Id of coupon to apply.
long taxonomyId	Id of taxonomy to apply coupon to.
bool includeTaxonomy	if true, include txonomy in coupon application. Else excludes taxonomy from coupon.

CouponApi.ApplyCouponToBasket Method

Applies a coupon to a customer's basket.

C#

```
public void ApplyCouponToBasket(long couponId, long basketId);
```

Visual Basic

```
Public Function ApplyCouponToBasket(couponId As long, basketId As long) As void
```

Parameters

Parameters	Description
long couponId	ID of coupon to apply to basket.
long basketId	ID of basket to apply coupon to.

CalculateBasketCoupons Method

This is the overview for the CalculateBasketCoupons method overload.

Overload List

	Name	Description
	CouponApi.CalculateBasketCoupons (Basket) (see page 623)	Runs coupon calculations on basket and returns BasketCalculatorData (see page 1442) object.
	CouponApi.CalculateBasketCoupons (long) (see page 623)	Runs coupon calculations on basket and returns BasketCalculatorData (see page 1442) object.

CouponApi.CalculateBasketCoupons Method (Basket)

Runs coupon calculations on basket and returns BasketCalculatorData (see page 1442) object.

C#

```
public BasketCalculatorData CalculateBasketCoupons(Basket basket);
```

Visual Basic

```
Public Function CalculateBasketCoupons(basket As Basket) As BasketCalculatorData
```

Parameters

Parameters	Description
Basket basket	Basket (see page 1437) class to calculate.

Returns

BasketCalculatorData (see page 1442) with coupon calculations complete.

CouponApi.CalculateBasketCoupons Method (long)

Runs coupon calculations on basket and returns BasketCalculatorData (see page 1442) object.

C#

```
public BasketCalculatorData CalculateBasketCoupons(long basketId);
```

Visual Basic

```
Public Function CalculateBasketCoupons(basketId As long) As BasketCalculatorData
```

Parameters

Parameters	Description
long basketId	Id of basket to calculate.

Returns

BasketCalculatorData (see page 1442) with coupon calculations complete.

CouponApi.Deactivate Method

Deactivate a coupondata object.

C#

```
public void Deactivate(long couponId);
```

Visual Basic

```
Public Function Deactivate(couponId As long) As void
```

Parameters

Parameters	Description
long couponId	Id of couponData to deactivate

CouponApi.Delete Method

Deletes a coupondata object.

C#

```
public void Delete(long couponId);
```

Visual Basic

```
Public Function Delete(couponId As long) As void
```

Parameters

Parameters	Description
long couponId	Id of couponData to delete

CouponApi.DeleteCouponProducts Method

Deletes all the products a coupon is applied to accept the products included in the ExclusionList.

C#

```
public void DeleteCouponProducts(long couponId, IEnumerable<long> exclusionIdList);
```

Visual Basic

```
Public Function DeleteCouponProducts(couponId As long, exclusionIdList As IEnumerable<long>)
As void
```

Parameters

Parameters	Description
long couponId	Id of coupon to remove from entries.
IEnumerable<long> exclusionIdList	List if entry Ids to exclude.

CouponApi.GetCatalogList Method

retrieves list of CouponEntryData (see page 1470) for catalogs with coupons applied to them.

C#

```
public List<CouponEntryData> GetCatalogList(Criteria<CouponEntryProperty> criteria);
```

Visual Basic

```
Public Function GetCatalogList(criteria As Criteria<CouponEntryProperty>) As
List<CouponEntryData>
```

Parameters

Parameters	Description
Criteria<CouponEntryProperty> criteria	criteria for formulating coupon retrieval

CouponApi.GetCouponsAppliedToBasket Method

Retrievies a list of coupons applied to a basket.

C#

```
public List<CouponData> GetCouponsAppliedToBasket(long basketId);
```

Visual Basic

```
Public Function GetCouponsAppliedToBasket(basketId As long) As List<CouponData>
```

Parameters

Parameters	Description
long basketId	Id of basket to retrieve coupons for.

Returns

List of CouponData (see page 1466) objects.

CouponApi.GetItem Method

Retrieves a CouponData (see page 1466) object by Id.

C#

```
public CouponData GetItem(long couponId);
```

Visual Basic

```
Public Function GetItem(couponId As long) As CouponData
```

Parameters

Parameters	Description
long couponId	Id of coupondata object to retrieve.

CouponApi.GetList Method

Retrieves list of couponData objects for a given customer.

C#

```
public List<CouponData> GetList(Criteria<CouponProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<CouponProperty>) As List<CouponData>
```

Parameters

Parameters	Description
customerId	Id of customer to retrieve coupones for.

GetProductList Method

This is the overview for the GetProductList method overload.

Overload List

	Name	Description
	CouponApi.GetProductList (Criteria<CouponEntryProperty>) (see page 626)	retrieves list of ProductCouponEntryData (see page 1574) for products with coupons available.
	CouponApi.GetProductList (long) (see page 626)	retrieves list of ProductCouponEntryData (see page 1574) for each product the coupon applies to.

CouponApi.GetProductList Method (Criteria<CouponEntryProperty>)

retrieves list of ProductCouponEntryData (see page 1574) for products with coupons available.

C#

```
public List<ProductCouponEntryData> GetProductList(Criteria<CouponEntryProperty> criteria);
```

Visual Basic

```
Public Function GetProductList(criteria As Criteria<CouponEntryProperty>) As
List<ProductCouponEntryData>
```

Parameters

Parameters	Description
Criteria<CouponEntryProperty> criteria	criteria for formulating product retrieval

CouponApi.GetProductList Method (long)

retrieves list of ProductCouponEntryData (see page 1574) for each product the coupon applies to.

C#

```
public List<ProductCouponEntryData> GetProductList(long couponId);
```

Visual Basic

```
Public Function GetProductList(couponId As long) As List<ProductCouponEntryData>
```

Parameters

Parameters	Description
long couponId	Id of coupon to search products for.

CouponApi.GetTaxonomyList Method

retrieves list of CouponEntryData (see page 1470) for taxonomies with coupons applied to them.

C#

```
public List<CouponEntryData> GetTaxonomyList(Criteria<CouponEntryProperty> criteria);
```

Visual Basic

```
Public Function GetTaxonomyList(criteria As Criteria<CouponEntryProperty>) As
List<CouponEntryData>
```

Parameters

Parameters	Description
Criteria<CouponEntryProperty> criteria	criteria for formulating coupon retrieval

CouponApi.IsCouponAppliedToProduct Method

Returns true if the coupon is applied to the catalog entry.

C#

```
public bool IsCouponAppliedToProduct(long couponId, long entryId);
```

Visual Basic

```
Public Function IsCouponAppliedToProduct(couponId As long, entryId As long) As bool
```

Parameters

Parameters	Description
long couponId	Id of coupon to to check.
long entryId	Id of entry to check for coupon.

CouponApi.IsCouponUsedForBasket Method

Checks if coupon has been applied to a basket.

C#

```
public bool IsCouponUsedForBasket(long couponId);
```

Visual Basic

```
Public Function IsCouponUsedForBasket(couponId As long) As bool
```

Parameters

Parameters	Description
long couponId	Id of coupon.

CouponApi.IsCouponUsedForOrder Method

Checks if coupon has been used by an order.

C#

```
public bool IsCouponUsedForOrder(long couponId);
```

Visual Basic

```
Public Function IsCouponUsedForOrder(couponId As long) As bool
```

Parameters

Parameters	Description
long couponId	Id of coupon.

CouponApi.SaveCouponApplications Method

Saves all the applications (taxonomy, catalog, product) for a given coupon. This is assumed to be an all inclusive lists. Any existing coupon applications that are not included will be deleted.

C#

```
public void SaveCouponApplications(long couponId, List<CouponEntryData> couponEntryList);
```

Visual Basic

```
Public Function SaveCouponApplications(couponId As long, couponEntryList As
List<CouponEntryData>) As void
```

Parameters

Parameters	Description
long couponId	Id of coupon to update.
List<CouponEntryData> couponEntryList	List of all CouponEntry applications for the given coupon.

CouponApi.Update Method

Updates the supplied CouponData (see page 1466) object.

C#

```
public void Update(CouponData couponData);
```

Visual Basic

```
Public Function Update(couponData As CouponData) As void
```

Parameters

Parameters	Description
CouponData couponData	CouponData (see page 1466) object to update.

ValidateCoupon Method

This is the overview for the ValidateCoupon method overload.

Overload List

	Name	Description
	CouponApi.ValidateCoupon (long) (see page 628)	Validates that the coupon has not expired and is still available for use.
	CouponApi.ValidateCoupon (long, long) (see page 628)	Validates that the coupon has not expired and is still available for use and can be used on the supplied product.

CouponApi.ValidateCoupon Method (long)

Validates that the coupon has not expired and is still available for use.

C#

```
public bool ValidateCoupon(long couponId);
```

Visual Basic

```
Public Function ValidateCoupon(couponId As long) As bool
```

Parameters

Parameters	Description
long couponId	Id of coupon to validate.

CouponApi.ValidateCoupon Method (long, long)

Validates that the coupon has not expired and is still available for use and can be used on the supplied product.

C#

```
public bool ValidateCoupon(long couponId, long productId);
```

Visual Basic

```
Public Function ValidateCoupon(couponId As long, productId As long) As bool
```

Parameters

Parameters	Description
long couponId	Id of coupon to validate.
long productId	Id of Product (see page 431) to validate for use with coupon.

CurrencyApi Class

Api class for managing commerce currencies in Cms (see page 7).

Class Hierarchy

C#

```
public class CurrencyApi : CommonApiBase;
```

Visual Basic

```
Public Class CurrencyApi
Inherits CommonApiBase
```

File

CurrencyApi.cs

CurrencyApi Methods

The methods of the CurrencyApi class are listed here.

Public Methods

	Name	Description
	Add (see page 629)	Adds a new Currency to Cms (see page 7). The Currency.Id property will be populated with the new Currency's Id.
	CanDelete (see page 629)	Checks to see if acurrency can be deleted.
	Delete (see page 630)	Deletes a Currency from the Cms (see page 7).
	GetItem (see page 630)	Retrieves a currency by currency Id.
	GetList (see page 630)	Retrieves a list of currencies.
	Update (see page 630)	Updates an existing Currency in Cms (see page 7).

CurrencyApi.Add Method

Adds a new Currency to Cms (see page 7). The Currency.Id property will be populated with the new Currency's Id.

C#

```
public void Add(CurrencyData currencyData);
```

Visual Basic

```
Public Function Add(currencyData As CurrencyData) As void
```

Parameters

Parameters	Description
CurrencyData currencyData	CurrencyData (see page 1472) object to add.

CurrencyApi.CanDelete Method

Checks to see if acurrency can be deleted.

C#

```
public bool CanDelete(int currencyId, out ValidationResults result);
```

Visual Basic

```
Public Function CanDelete(currencyId As Integer, result As out ValidationResults) As bool
```

Parameters

Parameters	Description
int currencyId	Id of currency to delete.
out ValidationResults result	ValidationResult returned containing errors if currency cannot be deleted.

Returns

true if currency can safely be deleted.

CurrencyApi.Delete Method

Deletes a Currency from the Cms (see page 7).

C#

```
public void Delete(int id);
```

Visual Basic

```
Public Function Delete(id As Integer) As void
```

Parameters

Parameters	Description
int id	Id of Currency to delete.

CurrencyApi.GetItem Method

Retrieves a currency by currency Id.

C#

```
public CurrencyData GetItem(int id);
```

Visual Basic

```
Public Function GetItem(id As Integer) As CurrencyData
```

Parameters

Parameters	Description
int id	Id of currency to retrieve.

CurrencyApi.GetList Method

Retrieves a list of currencies.

C#

```
public List<CurrencyData> GetList(Criteria<CurrencyProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<CurrencyProperty>) As List<CurrencyData>
```

Parameters

Parameters	Description
Criteria<CurrencyProperty> criteria	Criteria by which to filter currency being retrieved.

Returns

List of currencies meeting criteria filters.

CurrencyApi.Update Method

Updates an existing Currency in Cms (see page 7).

C#

```
public void Update(CurrencyData currencyData);
```

Visual Basic

```
Public Function Update(currencyData As CurrencyData) As void
```

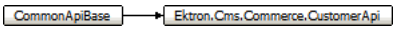
Parameters

Parameters	Description
CurrencyData currencyData	CurrencyData (see page 1472) object to save.

CustomerApi Class

Api class for managing Commerce (see page 1423) Customers in Cms (see page 7).

Class Hierarchy



C#

```
public class CustomerApi : CommonApiBase;
```

Visual Basic

```
Public Class CustomerApi
Inherits CommonApiBase
```

File

CustomerApi.cs

CustomerApi Methods

The methods of the CustomerApi class are listed here.

Public Methods

	Name	Description
	Add (see page 631)	This is the overview for the Add method overload.
	ChangeBillingAddress (see page 632)	Changes a customers billing address.
	ChangeShippingAddress (see page 633)	Changes a customers shipping address.
	Delete (see page 633)	Deletes a customer from the Cms (see page 7).
	GetItem (see page 633)	Retrieves a customer.
	GetList (see page 633)	Returns a list of customers based upon supplied criteria.
	GetOrderList (see page 634)	Returns the list of orders for a customer.
	GetOrderTotal (see page 634)	Returns the number of orders for a customer.
	Update (see page 634)	Updates an existing customer.

Add Method

This is the overview for the Add method overload.

Overload List

	Name	Description
	CustomerApi.Add (CustomerData, AddressData) (see page 632)	Adds a new customer with a billing address. The Customer.Id property will be populated with the new Customer's Id.
	CustomerApi.Add (CustomerData, AddressData, AddressData) (see page 632)	Adds a new customer with seperate billing and shipping addresses. The Customer.Id property will be populated with the new Customer's Id.

CustomerApi.Add Method (CustomerData, AddressData)

Adds a new customer with a billing address. The Customer.Id property will be populated with the new Customer's Id.

C#

```
public void Add(CustomerData customerData, AddressData billingAddress);
```

Visual Basic

```
Public Function Add(customerData As CustomerData, billingAddress As AddressData) As void
```

Parameters

Parameters	Description
CustomerData customerData	customer data to save.
AddressData billingAddress	Billing address of customer. Also serves as shipping address.

CustomerApi.Add Method (CustomerData, AddressData, AddressData)

Adds a new customer with seperate billing and shipping addresses. The Customer.Id property will be populated with the new Customer's Id.

C#

```
public void Add(CustomerData customerData, AddressData billingAddress, AddressData
shippingAddress);
```

Visual Basic

```
Public Function Add(customerData As CustomerData, billingAddress As AddressData,
shippingAddress As AddressData) As void
```

Parameters

Parameters	Description
CustomerData customerData	customer data to save.
AddressData billingAddress	Billing address of customer.
AddressData shippingAddress	Shipping address of customer.

CustomerApi.ChangeBillingAddress Method

Changes a customers billing address.

C#

```
public void ChangeBillingAddress(long customerId, long addressId);
```

Visual Basic

```
Public Function ChangeBillingAddress(customerId As long, addressId As long) As void
```

Parameters

Parameters	Description
long customerId	Id of cusotmer to delete.

CustomerApi.ChangeShippingAddress Method

Changes a customers shipping address.

C#

```
public void ChangeShippingAddress(long customerId, long addressId);
```

Visual Basic

```
Public Function ChangeShippingAddress(customerId As long, addressId As long) As void
```

Parameters

Parameters	Description
long customerId	Id of cusotmer to delete.

CustomerApi.Delete Method

Deletes a customer from the Cms (see page 7).

C#

```
public void Delete(long customerId);
```

Visual Basic

```
Public Function Delete(customerId As long) As void
```

Parameters

Parameters	Description
long customerId	Id of cusotmer to delete.

CustomerApi.GetItem Method

Retrieves a customer.

C#

```
public CustomerData GetItem(long customerId);
```

Visual Basic

```
Public Function GetItem(customerId As long) As CustomerData
```

Parameters

Parameters	Description
long customerId	Id of customer to retrieve.

CustomerApi.GetList Method

Returns a list of customers based upon supplied criteria.

C#

```
public List<CustomerData> GetList(Criteria<CustomerProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<CustomerProperty>) As List<CustomerData>
```

Parameters

Parameters	Description
Criteria<CustomerProperty> criteria	criteria by which to retrieve customer objects.

Returns

List of CustomerData (see page 1478)

CustomerApi.GetOrderList Method

Returns the list of orders for a customer.

C#

```
public List<OrderData> GetOrderList(long customerId);
```

Visual Basic

```
Public Function GetOrderList(customerId As long) As List<OrderData>
```

Parameters

Parameters	Description
long customerId	Id of customer to retrieve orders for.

CustomerApi.GetOrderTotal Method

Returns the number of orders for a customer.

C#

```
public long GetOrderTotal(long customerId);
```

Visual Basic

```
Public Function GetOrderTotal(customerId As long) As long
```

Parameters

Parameters	Description
long customerId	Id of customer to get order total for.

CustomerApi.Update Method

Updates an existing customer.

C#

```
public void Update(CustomerData customerData);
```

Visual Basic

```
Public Function Update(customerData As CustomerData) As void
```

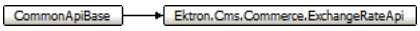
Parameters

Parameters	Description
CustomerData customerData	customer data to save.

ExchangeRateApi Class

Api class for managing commerce exchange rates in Cms (see page 7).

Class Hierarchy



C#

```
public class ExchangeRateApi : CommonApiBase;
```

Visual Basic

```
Public Class ExchangeRateApi
Inherits CommonApiBase
```

File

ExchangeRateApi.cs

ExchangeRateApi Methods

The methods of the ExchangeRateApi class are listed here.

Public Methods

	Name	Description
	Add (see page 635)	Adds a new Exchange Rate to the Cms (see page 7). The ExchangeRate.Id property will be populated with the new ExchangeRate's Id.
	ConvertPrice (see page 635)	Converts a price from the base currency to the exchange currency.
	Delete (see page 636)	Deletes an Exchange Rate from the Cms (see page 7).
	GetCurrentExchangeRate (see page 636)	Retrieves the current exchange rate by exchange currency Id.
	GetCurrentList (see page 636)	Retrieves a list of current exchange rates.
	GetList (see page 636)	Retrieves a list of exchange rates.
	Update (see page 637)	Updates an existing Exchange Rate in Cms (see page 7).

ExchangeRateApi.Add Method

Adds a new Exchange Rate to the Cms (see page 7). The ExchangeRate.Id property will be populated with the new ExchangeRate's Id.

C#

```
public void Add(ExchangeRateData exchangeRateData);
```

Visual Basic

```
Public Function Add(exchangeRateData As ExchangeRateData) As void
```

Parameters

Parameters	Description
ExchangeRateData exchangeRateData	ExchangeRateData (see page 1503) object to add.

ExchangeRateApi.ConvertPrice Method

Converts a price from the base currency to the exchange currency.

C#

```
public decimal ConvertPrice(decimal basePrice, int exchangeCurrencyId);
```

Visual Basic

```
Public Function ConvertPrice(basePrice As decimal, exchangeCurrencyId As Integer) As decimal
```

Parameters

Parameters	Description
decimal basePrice	price in the default currency.
int exchangeCurrencyId	

ExchangeRateApi.Delete Method

Deletes an Exchange Rate from the Cms (see page 7).

C#

```
public void Delete(ExchangeRateData exchangeRateData);
```

Visual Basic

```
Public Function Delete(exchangeRateData As ExchangeRateData) As void
```

Parameters

Parameters	Description
ExchangeRateData exchangeRateData	ExchangeRate to delete.

ExchangeRateApi.GetCurrentExchangeRate Method

Retrieves the current exchange rate by exchange currency Id.

C#

```
public ExchangeRateData GetCurrentExchangeRate(int exchangeCurrencyId);
```

Visual Basic

```
Public Function GetCurrentExchangeRate(exchangeCurrencyId As Integer) As ExchangeRateData
```

Parameters

Parameters	Description
CurrencyId	Id of exchangeRate to retrieve.

ExchangeRateApi.GetCurrentList Method

Retrieves a list of current exchange rates.

C#

```
public List<ExchangeRateData> GetCurrentList(Criteria<ExchangeRateProperty> criteria);
```

Visual Basic

```
Public Function GetCurrentList(criteria As Criteria<ExchangeRateProperty>) As
List<ExchangeRateData>
```

Parameters

Parameters	Description
Criteria<ExchangeRateProperty> criteria	Criteria by which to filter exchange rates being retrieved.

Returns

List of exchange rates meeting criteria filters.

ExchangeRateApi.GetList Method

Retrieves a list of exchange rates.

C#

```
public List<ExchangeRateData> GetList(Criteria<ExchangeRateProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<ExchangeRateProperty>) As List<ExchangeRateData>
```

Parameters

Parameters	Description
Criteria<ExchangeRateProperty> criteria	Criteria by which to filter exchange rates being retrieved.

Returns

List of exchange rates meeting criteria filters.

ExchangeRateApi.Update Method

Updates an existing Exchange Rate in Cms (see page 7).

C#

```
public void Update(ExchangeRateData exchangeRateData);
```

Visual Basic

```
Public Function Update(exchangeRateData As ExchangeRateData) As void
```

Parameters

Parameters	Description
ExchangeRateData exchangeRateData	ExchangeRateData (see page 1503) object to save.

InventoryApi Class

Api class for managing Commerce (see page 1423) Inventory (see page 1407) in Cms (see page 7).

Class Hierarchy

C#

```
public class InventoryApi : CommonApiBase;
```

Visual Basic

```
Public Class InventoryApi
Inherits CommonApiBase
```

File

InventoryApi.cs

InventoryApi Methods

The methods of the InventoryApi class are listed here.

Public Methods

	Name	Description
	SaveInventory (see page 637)	Saves inventory information for a catalog entry.

InventoryApi.SaveInventory Method

Saves inventory information for a catalog entry.

C#

```
public void SaveInventory(InventoryData inventory);
```

Visual Basic

```
Public Function SaveInventory(inventory As InventoryData) As void
```

Parameters

Parameters	Description
inventoryData	inventory to save.

OrderApi Class

Api class for managing Commerce (see page 1423) Orders in Cms (see page 7).

Class Hierarchy

C#

```
public class OrderApi : CommonApiBase;
```

Visual Basic

```
Public Class OrderApi
Inherits CommonApiBase
```

File

OrderApi.cs

OrderApi Methods

The methods of the OrderApi class are listed here.

Public Methods

	Name	Description
	Capture (see page 639)	This is the overview for the Capture method overload.
	GetCustomerOrderList (see page 639)	Retrieves a list of orders for a given customer.
	GetItem (see page 640)	Retrieves a single order.
	GetList (see page 640)	This is the overview for the GetList method overload.
	GetOnHoldOrderList (see page 641)	This is the overview for the GetOnHoldOrderList method overload.
	GetOrderMessage (see page 642)	This is the overview for the GetOrderMessage method overload.
	GetOrderPayment (see page 643)	Returns an order payment
	GetOrderStatus (see page 643)	Returns the status of an order.
	GetProductOrderList (see page 643)	This is the overview for the GetProductOrderList method overload.
	GetReport (see page 644)	Returns all the OrderReportData for given criteria.
	ItemsInStock (see page 644)	Returns true if all order items are in stock.
	MarkAsFraud (see page 645)	Marks an order as fraudulent.
	MarkPaymentAsSettled (see page 645)	This is the overview for the MarkPaymentAsSettled method overload.
	ProcessOrderInventory (see page 646)	Update all inventory levels for the supplied order.
	RollbackOrderInventory (see page 646)	Rolls back all inventory level updates for the supplied order.
	SetOrderStatus (see page 646)	Sets the status of an order.

	SetOrderToHold (see page 646)	Sets an order to OnHold.
	SetTrackingNumber (see page 647)	Sets the tracking number for an order part.

Capture Method

This is the overview for the Capture method overload.

Overload List

	Name	Description
	OrderApi.Capture (long) (see page 639)	Captures the payment for an order.
	OrderApi.Capture (long, long) (see page 639)	Captures the payment for an order.

OrderApi.Capture Method (long)

Captures the payment for an order.

C#

```
public void Capture(long orderId);
```

Visual Basic

```
Public Function Capture(orderId As long) As void
```

Parameters

Parameters	Description
long orderId	Id of order

OrderApi.Capture Method (long, long)

Captures the payment for an order.

C#

```
public void Capture(long orderId, long paymentId);
```

Visual Basic

```
Public Function Capture(orderId As long, paymentId As long) As void
```

Parameters

Parameters	Description
long orderId	Id of order

OrderApi.GetCustomerOrderList Method

Retrieves a list of orders for a given customer.

C#

```
public List<OrderData> GetCustomerOrderList(long customerId, PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetCustomerOrderList(customerId As long, pagingInfo As PagingInfo) As
List<OrderData>
```

Parameters

Parameters	Description
long customerId	Id of customer to retrieve orders for.
PagingInfo pagingInfo	paging information for retrieval.

Returns

List of OrderData (see page 1520) objects

OrderApi.GetItem Method

Retrieves a single order.

C#

```
public OrderData GetItem(long orderId);
```

Visual Basic

```
Public Function GetItem(orderId As long) As OrderData
```

Parameters

Parameters	Description
long orderId	Id of order to retrieve.

GetList Method

This is the overview for the GetList method overload.

Overload List

	Name	Description
	OrderApi.GetList (Criteria<OrderProperty>) (see page 640)	Retrieves a list of orders.
	OrderApi.GetList (DateTime, DateTime, PagingInfo) (see page 641)	Returns a list of orders for a given time period.

OrderApi.GetList Method (Criteria<OrderProperty>)

Retrieves a list of orders.

C#

```
public List<OrderData> GetList(Criteria<OrderProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<OrderProperty>) As List<OrderData>
```

Parameters

Parameters	Description
Criteria<OrderProperty> criteria	criteria by which to retrieve orders.

Returns

List of OrderData (see page 1520) objects

OrderApi.GetList Method (DateTime, DateTime, PagingInfo)

Returns a list of orders for a given time period.

C#

```
public List<OrderData> GetList(DateTime startDate, DateTime endDate, PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetList(startDate As DateTime, endDate As DateTime, pagingInfo As PagingInfo)
As List<OrderData>
```

Parameters

Parameters	Description
DateTime startDate	begininning of time period.
DateTime endDate	end of time period.
PagingInfo pagingInfo	paging information for list or orders.

Returns

List of OrderData (see page 1520) objects

GetOnHoldOrderList Method

This is the overview for the GetOnHoldOrderList method overload.

Overload List

	Name	Description
	OrderApi.GetOnHoldOrderList (PagingInfo) (see page 641)	Retrieves a list of orders on hold.
	OrderApi.GetOnHoldOrderList (PagingInfo, OrderProperty, EkEnumeration.OrderByDirection) (see page 641)	Retrieves a list of orders on hold.

OrderApi.GetOnHoldOrderList Method (PagingInfo)

Retrieves a list of orders on hold.

C#

```
public List<OrderData> GetOnHoldOrderList(PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetOnHoldOrderList(pagingInfo As PagingInfo) As List<OrderData>
```

Returns

List of OrderData (see page 1520) objects

List of OrderData (see page 1520) objects

OrderApi.GetOnHoldOrderList Method (PagingInfo, OrderProperty, EkEnumeration.OrderByDirection)

Retrieves a list of orders on hold.

C#

```
public List<OrderData> GetOnHoldOrderList(PagingInfo pagingInfo, OrderProperty orderBy,
EkEnumeration.OrderByDirection orderDirection);
```

Visual Basic

```
Public Function GetOnHoldOrderList(pagingInfo As PagingInfo, orderBy As OrderProperty,
orderDirection As EkEnumeration.OrderByDirection) As List<OrderData>
```

Parameters

Parameters	Description
OrderProperty orderBy	parameter to order list by.
EkEnumeration.OrderByDirection orderDirection	direction to order list in.

Returns

List of OrderData (see page 1520) objects

List of OrderData (see page 1520) objects

GetOrderMessage Method

This is the overview for the GetOrderMessage method overload.

Overload List

	Name	Description
	OrderApi.GetOrderMessage (EkEnumeration.EmailMessageTypes, long) (see page 642)	Returns the default EmailMessage for the supplied type.. Any tokens will be replaced based upon the supplied OrderId.
	OrderApi.GetOrderMessage (long, long) (see page 642)	Returns the EmailMessage for the supplied messageId. Any tokens will be replaced based upon the supplied OrderId.

OrderApi.GetOrderMessage Method (EkEnumeration.EmailMessageTypes, long)

Returns the default EmailMessage for the supplied type.. Any tokens will be replaced based upon the supplied OrderId.

C#

```
public EmailMessageData GetOrderMessage(EkEnumeration.EmailMessageTypes messageType, long
orderId);
```

Visual Basic

```
Public Function GetOrderMessage(messageType As EkEnumeration.EmailMessageTypes, orderId As
long) As EmailMessageData
```

Parameters

Parameters	Description
EkEnumeration.EmailMessageTypes messageType	Type of message to retrieve. i.e. OrderReceived, OrderShipped, OrderCancelled
long orderId	Id of order to retrieve message for.

OrderApi.GetOrderMessage Method (long, long)

Returns the EmailMessage for the supplied messageId. Any tokens will be replaced based upon the supplied OrderId.

C#

```
public EmailMessageData GetOrderMessage(long messageId, long orderId);
```

Visual Basic

```
Public Function GetOrderMessage(messageId As long, orderId As long) As EmailMessageData
```

Parameters

Parameters	Description
long messageId	Id of message to retrieve.
long orderId	Id of order to retrieve message for..

OrderApi.GetOrderPayment Method

Returns an order payment

C#

```
public OrderPaymentData GetOrderPayment(long paymentId);
```

Visual Basic

```
Public Function GetOrderPayment(paymentId As long) As OrderPaymentData
```

Parameters

Parameters	Description
long paymentId	The id of the payment to retrieve.

OrderApi.GetOrderStatus Method

Returns the status of an order.

C#

```
public EkEnumeration.OrderStatus GetOrderStatus(long orderId);
```

Visual Basic

```
Public Function GetOrderStatus(orderId As long) As EkEnumeration.OrderStatus
```

Parameters

Parameters	Description
long orderId	Id of order to retrieve status for.

GetProductOrderList Method

This is the overview for the GetProductOrderList method overload.

Overload List

	Name	Description
	OrderApi.GetProductOrderList (long, Criteria<OrderProperty>) (see page 643)	Retrieves a list of orders for a given product.
	OrderApi.GetProductOrderList (long, PagingInfo) (see page 644)	Retrieves a list of orders for a given product.

OrderApi.GetProductOrderList Method (long, Criteria<OrderProperty>)

Retrieves a list of orders for a given product.

C#

```
public List<OrderData> GetProductOrderList(long productId, Criteria<OrderProperty> criteria);
```

Visual Basic

```
Public Function GetProductOrderList(productId As long, criteria As Criteria<OrderProperty>) As
List<OrderData>
```

Parameters

Parameters	Description
long productId	Id of proiduct to retrieve orders for.
Criteria<OrderProperty> criteria	criteria information for filtering and sorting retrieval.

Returns

List of OrderData (see page 1520) objects

OrderApi.GetProductOrderList Method (long, PagingInfo)

Retrieves a list of orders for a given product.

C#

```
public List<OrderData> GetProductOrderList(long productId, PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetProductOrderList(productId As long, pagingInfo As PagingInfo) As
List<OrderData>
```

Parameters

Parameters	Description
long productId	Id of proiduct to retrieve orders for.
PagingInfo pagingInfo	paging information for retrieval.

Returns

List of OrderData (see page 1520) objects

OrderApi.GetReport Method

Returns all the OrderReportData for given criteria.

C#

```
public Reporting.OrderReportData GetReport(Criteria<OrderProperty> criteria);
```

Visual Basic

```
Public Function GetReport(criteria As Criteria<OrderProperty>) As Reporting.OrderReportData
```

Parameters

Parameters	Description
Criteria<OrderProperty> criteria	criteria for reporting.

OrderApi.ItemsInStock Method

Returns true if all order items are in stock.

C#

```
public bool ItemsInStock(long orderId);
```

Visual Basic

```
Public Function ItemsInStock(orderId As long) As bool
```

Parameters

Parameters	Description
long orderId	Id of order to verify stock levels for.

OrderApi.MarkAsFraud Method

Marks an order as fraudulent.

C#

```
public void MarkAsFraud(long orderId);
```

Visual Basic

```
Public Function MarkAsFraud(orderId As long) As void
```

Parameters

Parameters	Description
long orderId	Id of order

MarkPaymentAsSettled Method

This is the overview for the MarkPaymentAsSettled method overload.

Overload List

	Name	Description
	OrderApi.MarkPaymentAsSettled (long) (see page 645)	Marks a payment as settled.
	OrderApi.MarkPaymentAsSettled (long, long) (see page 645)	Marks a payment as settled.

OrderApi.MarkPaymentAsSettled Method (long)

Marks a payment as settled.

C#

```
public void MarkPaymentAsSettled(long orderId);
```

Visual Basic

```
Public Function MarkPaymentAsSettled(orderId As long) As void
```

Parameters

Parameters	Description
long orderId	Id of order

OrderApi.MarkPaymentAsSettled Method (long, long)

Marks a payment as settled.

C#

```
public void MarkPaymentAsSettled(long orderId, long paymentId);
```

Visual Basic

```
Public Function MarkPaymentAsSettled(orderId As long, paymentId As long) As void
```

Parameters

Parameters	Description
long orderId	Id of order

OrderApi.ProcessOrderInventory Method

Update all inventory levels for the supplied order.

C#

```
public void ProcessOrderInventory(long orderId);
```

Visual Basic

```
Public Function ProcessOrderInventory(orderId As long) As void
```

Parameters

Parameters	Description
long orderId	Id of order to process.

OrderApi.RollbackOrderInventory Method

Rolls back all inventory level updates for the supplied order.

C#

```
public void RollbackOrderInventory(long orderId);
```

Visual Basic

```
Public Function RollbackOrderInventory(orderId As long) As void
```

Parameters

Parameters	Description
long orderId	Id of order to rollback.

OrderApi.SetOrderStatus Method

Sets the status of an order.

C#

```
public void SetOrderStatus(long orderId, EkEnumeration.OrderStatus status);
```

Visual Basic

```
Public Function SetOrderStatus(orderId As long, status As EkEnumeration.OrderStatus) As void
```

Parameters

Parameters	Description
long orderId	Id of order
EkEnumeration.OrderStatus status	Status to change the order to.

OrderApi.SetOrderToHold Method

Sets an order to OnHold.

C#

```
public void SetOrderToHold(long orderId);
```

Visual Basic

```
Public Function SetOrderToHold(orderId As long) As void
```

Parameters

Parameters	Description
long orderId	Id of order

OrderApi.SetTrackingNumber Method

Sets the tracking number for an order part.

C#

```
public void SetTrackingNumber(long orderId, string trackingNumber, bool markAsShipped);
```

Visual Basic

```
Public Function SetTrackingNumber(orderId As long, trackingNumber As string, markAsShipped As
bool) As void
```

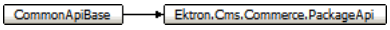
Parameters

Parameters	Description
long orderId	Id of order to set tracking number.
string trackingNumber	The tracking number
bool markAsShipped	mark the item as shipped.

PackageApi Class

API (see page 669) class used for managing shipping packages.

Class Hierarchy



C#

```
public class PackageApi : CommonApiBase;
```

Visual Basic

```
Public Class PackageApi
Inherits CommonApiBase
```

File

PackageApi.cs

PackageApi Methods

The methods of the PackageApi class are listed here.

Public Methods

	Name	Description
	Add (see page 647)	Adds a new Package to Cms (see page 7). The Package.Id property will be populated with the new Package's Id.
	Delete (see page 648)	Deletes a Package from the Cms (see page 7).
	GetItem (see page 648)	Retrieves a package by package Id.
	GetList (see page 648)	Retrieves a list of Packages.
	Update (see page 649)	Updates an existing Package in Cms (see page 7).

PackageApi.Add Method

Adds a new Package to Cms (see page 7). The Package.Id property will be populated with the new Package's Id.

C#

```
public void Add(PackageData packageData);
```

Visual Basic

```
Public Function Add(packageData As PackageData) As void
```

Parameters

Parameters	Description
PackageData packageData	PackageData (see page 1532) object to add.

PackageApi.Delete Method

Deletes a Package from the Cms (see page 7).

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	Id of Package to delete.

PackageApi.GetItem Method

Retrieves a package by package Id.

C#

```
public PackageData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As PackageData
```

Parameters

Parameters	Description
long id	Id of package to retrieve.

PackageApi.GetList Method

Retrieves a list of Packages.

C#

```
public List<PackageData> GetList(Criteria<PackageProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<PackageProperty>) As List<PackageData>
```

Parameters

Parameters	Description
Criteria<PackageProperty> criteria	Criteria by which to filter package being retrieved.

Returns

List of Packages meeting criteria filters.

PackageApi.Update Method

Updates an existing Package in Cms (see page 7).

C#

```
public void Update(PackageData packageData);
```

Visual Basic

```
Public Function Update(packageData As PackageData) As void
```

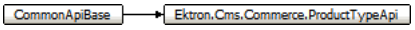
Parameters

Parameters	Description
PackageData packageData	PackageData (see page 1532) object to save.

ProductTypeApi Class

Api class for managing Commerce (see page 1423) product types in Cms (see page 7).

Class Hierarchy



C#

```
public class ProductTypeApi : CommonApiBase;
```

Visual Basic

```
Public Class ProductTypeApi
Inherits CommonApiBase
```

File

ProductTypeApi.cs

ProductTypeApi Methods

The methods of the ProductTypeApi class are listed here.

Public Methods

	Name	Description
	Add (see page 650)	This is Add, a member of class ProductTypeApi.
	AddProductAttribute (see page 650)	This is AddProductAttribute, a member of class ProductTypeApi.
	Delete (see page 650)	This is Delete, a member of class ProductTypeApi.
	DeleteProductAttribute (see page 650)	This is the overview for the DeleteProductAttribute method overload.
	GetFolderProductTypeList (see page 651)	This is GetFolderProductTypeList, a member of class ProductTypeApi.
	GetItem (see page 651)	This is the overview for the GetItem method overload.
	GetProductAttribute (see page 652)	This is GetProductAttribute, a member of class ProductTypeApi.
	GetProductAttributeList (see page 652)	This is GetProductAttributeList, a member of class ProductTypeApi.
	Update (see page 652)	This is Update, a member of class ProductTypeApi.
	UpdateProductAttribute (see page 652)	This is UpdateProductAttribute, a member of class ProductTypeApi.

ProductTypeApi.Add Method

C#

```
public void Add(ProductTypeData productType);
```

Visual Basic

```
Public Function Add(productType As ProductTypeData) As void
```

Description

This is Add, a member of class ProductTypeApi.

ProductTypeApi.AddProductAttribute Method

C#

```
public long AddProductAttribute(long productTypeId, ProductTypeAttributeData attribute,
System.Data.IDbTransaction transaction);
```

Visual Basic

```
Public Function AddProductAttribute(productTypeId As long, attribute As
ProductTypeAttributeData, transaction As System.Data.IDbTransaction) As long
```

Description

This is AddProductAttribute, a member of class ProductTypeApi.

ProductTypeApi.Delete Method

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Description

This is Delete, a member of class ProductTypeApi.

DeleteProductAttribute Method

This is the overview for the DeleteProductAttribute method overload.

Overload List

	Name	Description
	ProductTypeApi.DeleteProductAttribute (long, bool, System.Data.IDbTransaction) (see page 651)	This is DeleteProductAttribute, a member of class ProductTypeApi.
	ProductTypeApi.DeleteProductAttribute (long, System.Data.IDbTransaction) (see page 650)	This is DeleteProductAttribute, a member of class ProductTypeApi.

ProductTypeApi.DeleteProductAttribute Method (long, System.Data.IDbTransaction)

C#

```
public void DeleteProductAttribute(long attributeId, System.Data.IDbTransaction transaction);
```

Visual Basic

```
Public Function DeleteProductAttribute(attributeId As long, transaction As
System.Data.IDbTransaction) As void
```

Description

This is DeleteProductAttribute, a member of class ProductTypeApi.

ProductTypeApi.DeleteProductAttribute Method (long, bool, System.Data.IDbTransaction)

C#

```
public void DeleteProductAttribute(long attributeId, bool forceDelete,
System.Data.IDbTransaction transaction);
```

Visual Basic

```
Public Function DeleteProductAttribute(attributeId As long, forceDelete As bool, transaction
As System.Data.IDbTransaction) As void
```

Description

This is DeleteProductAttribute, a member of class ProductTypeApi.

ProductTypeApi.GetFolderProductTypeList Method

C#

```
public List<ProductTypeData> GetFolderProductTypeList(long folderId);
```

Visual Basic

```
Public Function GetFolderProductTypeList(folderId As long) As List<ProductTypeData>
```

Description

This is GetFolderProductTypeList, a member of class ProductTypeApi.

GetItem Method

This is the overview for the GetItem method overload.

Overload List

	Name	Description
	ProductTypeApi.GetItem (long) (see page 651)	This is GetItem, a member of class ProductTypeApi.
	ProductTypeApi.GetItem (long, bool) (see page 652)	This is GetItem, a member of class ProductTypeApi.

ProductTypeApi.GetItem Method (long)

C#

```
public ProductTypeData GetItem(long productTypeId);
```

Visual Basic

```
Public Function GetItem(productTypeId As long) As ProductTypeData
```

Description

This is GetItem, a member of class ProductTypeApi.

ProductTypeApi.GetItem Method (long, bool)

C#

```
public ProductTypeData GetItem(long productTypeId, bool withAttributes);
```

Visual Basic

```
Public Function GetItem(productTypeId As long, withAttributes As bool) As ProductTypeData
```

Description

This is GetItem, a member of class ProductTypeApi.

ProductTypeApi.GetProductAttribute Method

C#

```
public ProductTypeAttributeData GetProductAttribute(long attributeId);
```

Visual Basic

```
Public Function GetProductAttribute(attributeId As long) As ProductTypeAttributeData
```

Description

This is GetProductAttribute, a member of class ProductTypeApi.

ProductTypeApi.GetProductAttributeList Method

C#

```
public List<ProductTypeAttributeData> GetProductAttributeList(long productTypeId);
```

Visual Basic

```
Public Function GetProductAttributeList(productTypeId As long) As
List<ProductTypeAttributeData>
```

Description

This is GetProductAttributeList, a member of class ProductTypeApi.

ProductTypeApi.Update Method

C#

```
public void Update(ProductTypeData productType);
```

Visual Basic

```
Public Function Update(productType As ProductTypeData) As void
```

Description

This is Update, a member of class ProductTypeApi.

ProductTypeApi.UpdateProductAttribute Method

C#

```
public void UpdateProductAttribute(ProductTypeAttributeData attribute,
System.Data.IDbTransaction transaction);
```

Visual Basic

```
Public Function UpdateProductAttribute(attribute As ProductTypeAttributeData, transaction As
System.Data.IDbTransaction) As void
```

Description

This is UpdateProductAttribute, a member of class ProductTypeApi.

RecommendationApi Class

Api class for managing entry reccomendations.

Class Hierarchy

C#

```
public class RecommendationApi : CommonApiBase;
```

Visual Basic

```
Public Class RecommendationApi
Inherits CommonApiBase
```

File

RecommendationApi.cs

RecommendationApi Methods

The methods of the RecommendationApi class are listed here.

Public Methods

	Name	Description
	Add (see page 653)	Adds a new entry recommendation to cms. The supplied recommendation will be updated with the new Id.
	Delete (see page 653)	Deletes a recommendation.
	DeleteRecommendationByEntry (see page 654)	Delete (see page 653) recommendations associated with a catalog entry.
	GetItem (see page 654)	Retrieves a recommendation by id.
	Update (see page 654)	Updates an existing entry recommendation in cms.
	UpdateCrossSell (see page 654)	This is UpdateCrossSell, a member of class RecommendationApi.
	UpdateUpSell (see page 655)	This is UpdateUpSell, a member of class RecommendationApi.

RecommendationApi.Add Method

Adds a new entry recommendation to cms. The supplied recommendation will be updated with the new Id.

C#

```
public void Add(RecommendationItemData recommendation);
```

Visual Basic

```
Public Function Add(recommendation As RecommendationItemData) As void
```

Parameters

Parameters	Description
RecommendationItemData recommendation	recommendation data object to add.

RecommendationApi.Delete Method

Deletes a recommendation.

C#

```
public void Delete(long recommendationId);
```

Visual Basic

```
Public Function Delete(recommendationId As long) As void
```

Parameters

Parameters	Description
long recommendationId	id of recommendation to delete.

RecommendationApi.DeleteRecommendationByEntry Method

Delete (see page 653) recommendations associated with a catalog entry.

C#

```
public void DeleteRecommendationByEntry(long entryId);
```

Visual Basic

```
Public Function DeleteRecommendationByEntry(entryId As long) As void
```

Parameters

Parameters	Description
long entryId	Id of entry to delete recommendations for.

RecommendationApi.GetItem Method

Retrieves a recommendation by id.

C#

```
public RecommendationItemData GetItem(long recommendationId);
```

Visual Basic

```
Public Function GetItem(recommendationId As long) As RecommendationItemData
```

Parameters

Parameters	Description
long recommendationId	Id of recommendation to retrieve.

RecommendationApi.Update Method

Updates an existing entry recommendation in cms.

C#

```
public void Update(RecommendationItemData recommendation);
```

Visual Basic

```
Public Function Update(recommendation As RecommendationItemData) As void
```

Parameters

Parameters	Description
RecommendationItemData recommendation	recommendation data object to update.

RecommendationApi.UpdateCrossSell Method

C#

```
public void UpdateCrossSell(long entryId, IEnumerable<RecommendationItemData> recommendations);
```

Visual Basic

```
Public Function UpdateCrossSell(entryId As long, recommendations As
IEnumerable<RecommendationItemData>) As void
```

Description

This is UpdateCrossSell, a member of class RecommendationApi.

RecommendationApi.UpdateUpSell Method

C#

```
public void UpdateUpSell(long entryId, IEnumerable<RecommendationItemData> recommendations);
```

Visual Basic

```
Public Function UpdateUpSell(entryId As long, recommendations As
IEnumerable<RecommendationItemData>) As void
```

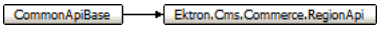
Description

This is UpdateUpSell, a member of class RecommendationApi.

RegionApi Class

Api class for managing regions in Cms (see page 7).

Class Hierarchy



C#

```
public class RegionApi : CommonApiBase;
```

Visual Basic

```
Public Class RegionApi
Inherits CommonApiBase
```

File

RegionApi.cs

RegionApi Methods

The methods of the RegionApi class are listed here.

Public Methods

	Name	Description
	Add (see page 655)	Adds a new Region to the Cms (see page 7). The Region.Id property will be populated with the new regions Id.
	CanDelete (see page 656)	Checks to see if a Region can be deleted.
	Delete (see page 656)	Deletes a region from the Cms (see page 7).
	GetItem (see page 656)	Retrieves a region by region Id.
	GetList (see page 657)	Retrieves a list of regions.
	Update (see page 657)	Updates an existing Region in the Cms (see page 7).

RegionApi.Add Method

Adds a new Region to the Cms (see page 7). The Region.Id property will be populated with the new regions Id.

C#

```
public void Add(RegionData regionData);
```

Visual Basic

```
Public Function Add(regionData As RegionData) As void
```

Parameters

Parameters	Description
RegionData regionData	RegionData (see page 1554) object to add.

RegionApi.CanDelete Method

Checks to see if a Region can be deleted.

C#

```
public bool CanDelete(long regionId, out ValidationResults result);
```

Visual Basic

```
Public Function CanDelete(regionId As long, result As out ValidationResults) As bool
```

Parameters

Parameters	Description
out ValidationResults result	ValidationResult returned containing messages if Region cannot be deleted.
currencyId	Id of Region to delete.

Returns

true if region can safely be deleted.

RegionApi.Delete Method

Deletes a region from the Cms (see page 7).

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	Id of region to delete.

RegionApi.GetItem Method

Retrieves a region by region Id.

C#

```
public RegionData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As RegionData
```

Parameters

Parameters	Description
long id	Id of region to retrieve.

RegionApi.GetList Method

Retrieves a list of regions.

C#

```
public List<RegionData> GetList(Criteria<RegionProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<RegionProperty>) As List<RegionData>
```

Parameters

Parameters	Description
Criteria<RegionProperty> criteria	Criteria by which to filter regions being retrieved.

Returns

List of regions meeting criteria filters.

RegionApi.Update Method

Updates an existing Region in the Cms (see page 7).

C#

```
public void Update(RegionData regionData);
```

Visual Basic

```
Public Function Update(regionData As RegionData) As void
```

Parameters

Parameters	Description
RegionData regionData	RegionData (see page 1554) object to save.

ShippingMethodApi Class

Api class for managing shipping methods in CMS.

Class Hierarchy

C#

```
public class ShippingMethodApi : CommonApiBase;
```

Visual Basic

```
Public Class ShippingMethodApi
Inherits CommonApiBase
```

File

ShippingApi.cs

ShippingMethodApi Methods

The methods of the ShippingMethodApi class are listed here.

Public Methods

	Name	Description
	Add (see page 658)	This is Add, a member of class ShippingMethodApi.

	Update (see page 658)	This is Update, a member of class ShippingMethodApi.

ShippingMethodApi.Add Method

C#

```
public void Add(ShippingMethodData shipOption);
```

Visual Basic

```
Public Function Add(shipOption As ShippingMethodData) As void
```

Description

This is Add, a member of class ShippingMethodApi.

ShippingMethodApi.Update Method

C#

```
public void Update(ShippingMethodData shipOption);
```

Visual Basic

```
Public Function Update(shipOption As ShippingMethodData) As void
```

Description

This is Update, a member of class ShippingMethodApi.

TaxApi Class

Api class for managing Commerce (see page 1423) Tax rates in Cms (see page 7).

Class Hierarchy

C#

```
public class TaxApi : CommonApiBase;
```

Visual Basic

```
Public Class TaxApi
Inherits CommonApiBase
```

File

TaxApi.cs

TaxApi Methods

The methods of the TaxApi class are listed here.

Public Methods

	Name	Description
	Add (see page 659)	Adds a new Tax rate to Cms (see page 7). The Tax.Id property will be populated with the new Tax's Id.
	Delete (see page 659)	Deletes a Tax rate from the Cms (see page 7).
V	GetApplicableTaxRate (see page 659)	Returns the applicable tax rate for the supplied addresses and tax class.
V	GetApplicableTaxRateList (see page 660)	Returns a list of applicable tax rates for the supplied addresses. There will be at most one rate for each available tax class for the the specified addresses.
	GetItem (see page 660)	This is the overview for the GetItem method overload.

	GetItemByCountryId (see page 661)	Retrieves a tax rate based upon tax class and country Id. For Example, retrieves a TaxRateType.CountrySalesTax with the country Id of 840..
	GetItemByPostalCode (see page 661)	This is the overview for the GetItemByPostalCode method overload.
	GetItemByPostalId (see page 662)	Retrieves a tax rate based upon tax class and postal code Id. For Example, retrieves a TaxRateType.PostalSalesTax with the Postal Code Id of 40..
	GetItemByRegionId (see page 662)	Retrieves a tax rate based upon tax class and Type Item Id. For Example, retrieves a TaxRateType.RegionSalesTax with the RegionId of 40..
	GetList (see page 662)	Retrieves a list of tax rates.
	Update (see page 663)	Updates an existing tax rate in Cms (see page 7).

TaxApi.Add Method

Adds a new Tax rate to Cms (see page 7). The Tax.Id property will be populated with the new Tax's Id.

C#

```
public void Add(TaxRateData taxRateData);
```

Visual Basic

```
Public Function Add(taxRateData As TaxRateData) As void
```

Parameters

Parameters	Description
TaxRateData taxRateData	TaxRateData (see page 1563) object to add.

TaxApi.Delete Method

Deletes a Tax rate from the Cms (see page 7).

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	Id of Tax to delete.

TaxApi.GetApplicableTaxRate Method

Returns the applicable tax rate for the supplied addresses and tax class.

C#

```
public virtual TaxRateData GetApplicableTaxRate(AddressData shipFromAddress, AddressData
shipToAddress, long taxClassId);
```

Visual Basic

```
Public virtual Function GetApplicableTaxRate(shipFromAddress As AddressData, shipToAddress As
AddressData, taxClassId As long) As TaxRateData
```

Parameters

Parameters	Description
AddressData shipFromAddress	Address where the products are coming from.
AddressData shipToAddress	Address where products are going.
long taxClassId	The taxClass for which to retrieve a rate

TaxApi.GetApplicableTaxRateList Method

Returns a list of applicable tax rates for the supplied addresses. There will be at most one rate for each available tax class for the the specified addresses.

C#

```
public virtual List<TaxRateData> GetApplicableTaxRateList(AddressData shipFromAddress,
AddressData shipToAddress);
```

Visual Basic

```
Public virtual Function GetApplicableTaxRateList(shipFromAddress As AddressData, shipToAddress
As AddressData) As List<TaxRateData>
```

Parameters

Parameters	Description
AddressData shipFromAddress	Address where the products are coming from.
AddressData shipToAddress	Address where products are going.

GetItem Method

This is the overview for the GetItem method overload.

Overload List

	Name	Description
	TaxApi.GetItem (long) (see page 660)	Retrieves a tax rate by Id.
	TaxApi.GetItem (long, long) (see page 660)	Retrieves a tax rate based up Tax Rate Type and Type Item Id. For Example, retrieves a TaxRateType.RegionSalesTax with the RegionId of 40..

TaxApi.GetItem Method (long)

Retrieves a tax rate by Id.

C#

```
public TaxRateData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As TaxRateData
```

Parameters

Parameters	Description
long id	Id of tax rate to retrieve.

TaxApi.GetItem Method (long, long)

Retrieves a tax rate based up Tax Rate Type and Type Item Id. For Example, retrieves a TaxRateType.RegionSalesTax with the RegionId of 40..

C#

```
public TaxRateData GetItem(long taxTypeId, long typeItemId);
```

Visual Basic

```
Public Function GetItem(taxTypeId As long, typeItemId As long) As TaxRateData
```

Parameters

Parameters	Description
long taxTypeId	Type of reate to retrieve.
long typeItemId	Id of corresponding object. For example, RegionId for TaxRateType.RegionSalesTax

TaxApi.GetItemByCountryId Method

Retrieves a tax rate based upon tax class and country Id. For Example, retrieves a TaxRateType.CountrySalesTax with the country Id of 840..

C#

```
public virtual TaxRateData GetItemByCountryId(long taxClassId, long countryId);
```

Visual Basic

```
Public virtual Function GetItemByCountryId(taxClassId As long, countryId As long) As
TaxRateData
```

Parameters

Parameters	Description
long taxClassId	Id of tax class to retrieve rate for.
long countryId	Id of country to retrieve tax for.

GetItemByPostalCode Method

This is the overview for the GetItemByPostalCode method overload.

Overload List

	Name	Description
	TaxApi.GetItemByPostalCode (long, string) (see page 661)	Retrieves a tax rate based upon tax class and postal code .
	TaxApi.GetItemByPostalCode (long, string, int) (see page 661)	Retrieves a tax rate based upon tax class and postal code.

TaxApi.GetItemByPostalCode Method (long, string)

Retrieves a tax rate based upon tax class and postal code .

C#

```
public TaxRateData GetItemByPostalCode(long taxClassId, string postalCode);
```

Visual Basic

```
Public Function GetItemByPostalCode(taxClassId As long, postalCode As string) As TaxRateData
```

Parameters

Parameters	Description
long taxClassId	Id of tax class to retrieve rate for.
string postalCode	postalCode to retrieve taxrate for.

TaxApi.GetItemByPostalCode Method (long, string, int)

Retrieves a tax rate based upon tax class and postal code.

C#

```
public TaxRateData GetItemByPostalCode(long taxClassId, string postalCode, int countryId);
```

Visual Basic

```
Public Function GetItemByPostalCode(taxClassId As long, postalCode As string, countryId As
Integer) As TaxRateData
```

Parameters

Parameters	Description
long taxClassId	Id of tax class to retrieve rate for.
string postalCode	postalCode to retrieve taxrate for.
int countryId	Id of country associated with PostalCode

TaxApi.GetItemByPostalId Method

Retrieves a tax rate based upon tax class and postal code Id. For Example, retrieves a TaxRateType.PostalSalesTax with the Postal Code Id of 40..

C#

```
public TaxRateData GetItemByPostalId(long taxClassId, long postalCodeId);
```

Visual Basic

```
Public Function GetItemByPostalId(taxClassId As long, postalCodeId As long) As TaxRateData
```

Parameters

Parameters	Description
long taxClassId	Id of tax class to retrieve rate for.
long postalCodeId	Id of postal code to retrieve tax for.

TaxApi.GetItemByRegionId Method

Retrieves a tax rate based upon tax class and Type Item Id. For Example, retrieves a TaxRateType.RegionSalesTax with the RegionId of 40..

C#

```
public TaxRateData GetItemByRegionId(long taxClassId, long regionId);
```

Visual Basic

```
Public Function GetItemByRegionId(taxClassId As long, regionId As long) As TaxRateData
```

Parameters

Parameters	Description
long taxClassId	Type of reate to retrieve.
long regionId	Id of Region to retrieve rate for.

TaxApi.GetList Method

Retrieves a list of tax rates.

C#

```
public List<TaxRateData> GetList(Criteria<TaxRateProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<TaxRateProperty>) As List<TaxRateData>
```

Parameters

Parameters	Description
Criteria<TaxRateProperty> criteria	Criteria by which to filter tax being retrieved.

Returns

List of countries meeting criteria filters.

TaxApi.Update Method

Updates an existing tax rate in Cms (see page 7).

C#

```
public void Update(TaxRateData taxRateData);
```

Visual Basic

```
Public Function Update(taxRateData As TaxRateData) As void
```

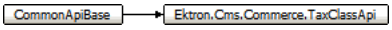
Parameters

Parameters	Description
TaxRateData taxRateData	TaxRateData (see page 1563) object to save.

TaxClassApi Class

Api class for managing Commerce (see page 1423) Tax Classes in Cms (see page 7).

Class Hierarchy



C#

```
public class TaxClassApi : CommonApiBase;
```

Visual Basic

```
Public Class TaxClassApi
Inherits CommonApiBase
```

File

TaxClassApi.cs

TaxClassApi Methods

The methods of the TaxClassApi class are listed here.

Public Methods

	Name	Description
	Add (see page 663)	Adds a new Tax Class to Cms (see page 7). The TaxClass.Id property will be populated with the new TaxClass's Id.
	Delete (see page 664)	Deletes a Tax Class from the Cms (see page 7).
	GetItem (see page 664)	Retrieves a tax class by Id.
	GetList (see page 664)	Retrieves a list of Tax Classes.
	IsUsed (see page 665)	Returns whether a TaxClass is used in a product or order.
	Update (see page 665)	Updates an existing Tax Class in Cms (see page 7).

TaxClassApi.Add Method

Adds a new Tax Class to Cms (see page 7). The TaxClass.Id property will be populated with the new TaxClass's Id.

C#

```
public void Add(TaxClassData taxClassData);
```

Visual Basic

```
Public Function Add(taxClassData As TaxClassData) As void
```

Parameters

Parameters	Description
TaxClassData taxClassData	TaxClassData (see page 1561) object to add.

TaxClassApi.Delete Method

Deletes a Tax Class from the Cms (see page 7).

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	Id of TaxClass to delete.

TaxClassApi.GetItem Method

Retrieves a tax class by Id.

C#

```
public TaxClassData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As TaxClassData
```

Parameters

Parameters	Description
long id	Id of taxClass to retrieve.

TaxClassApi.GetList Method

Retrieves a list of Tax Classes.

C#

```
public List<TaxClassData> GetList(Criteria<TaxClassProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<TaxClassProperty>) As List<TaxClassData>
```

Parameters

Parameters	Description
Criteria<TaxClassProperty> criteria	Criteria by which to filter taxClass being retrieved.

Returns

List of TaxClass data objects.

TaxClassApi.IsUsed Method

Returns whether a TaxClass is used in a product or order.

C#

```
public bool IsUsed(long id);
```

Visual Basic

```
Public Function IsUsed(id As long) As bool
```

Parameters

Parameters	Description
long id	Id of TaxClass.

TaxClassApi.Update Method

Updates an existing Tax Class in Cms (see page 7).

C#

```
public void Update(TaxClassData taxClassData);
```

Visual Basic

```
Public Function Update(taxClassData As TaxClassData) As void
```

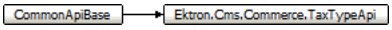
Parameters

Parameters	Description
TaxClassData taxClassData	TaxClassData (see page 1561) object to save.

TaxTypeApi Class

Api class for managing Commerce (see page 1423) tax types in Cms (see page 7).

Class Hierarchy



C#

```
public class TaxTypeApi : CommonApiBase;
```

Visual Basic

```
Public Class TaxTypeApi
Inherits CommonApiBase
```

File

TaxTypeApi.cs

TaxTypeApi Methods

The methods of the TaxTypeApi class are listed here.

Public Methods

	Name	Description
	Add (see page 666)	Adds a new Tax Type Cms (see page 7). The TaxType.Id property will be populated with the new TaxType's Id.
	Delete (see page 666)	Deletes a Tax Type from the Cms (see page 7).
	GetItem (see page 666)	Retrieves a tax type by Id.
	GetList (see page 666)	Retrieves a list of Tax Types.

	Update (see page 667)	Updates an existing Tax Type in Cms (see page 7).

TaxTypeApi.Add Method

Adds a new Tax Type Cms (see page 7). The TaxType.Id property will be populated with the new TaxType's Id.

C#

```
public void Add(TaxTypeData taxTypeData);
```

Visual Basic

```
Public Function Add(taxTypeData As TaxTypeData) As void
```

Parameters

Parameters	Description
TaxTypeData taxTypeData	TaxTypeData (see page 1565) object to add.

TaxTypeApi.Delete Method

Deletes a Tax Type from the Cms (see page 7).

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	Id of TaxType to delete.

TaxTypeApi.GetItem Method

Retrieves a tax type by Id.

C#

```
public TaxTypeData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As TaxTypeData
```

Parameters

Parameters	Description
long id	Id of taxType to retrieve.

TaxTypeApi.GetList Method

Retrieves a list of Tax Types.

C#

```
public List<TaxTypeData> GetList(Criteria<TaxTypeProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<TaxTypeProperty>) As List<TaxTypeData>
```

Parameters

Parameters	Description
Criteria<TaxTypeProperty> criteria	Criteria by which to filter taxType being retrieved.

Returns

List of TaxTypeData (see page 1565) objects.

TaxTypeApi.Update Method

Updates an existing Tax Type in Cms (see page 7).

C#

```
public void Update(TaxTypeData taxTypeData);
```

Visual Basic

```
Public Function Update(taxTypeData As TaxTypeData) As void
```

Parameters

Parameters	Description
TaxTypeData taxTypeData	TaxTypeData (see page 1565) object to save.

WarehouseApi Class

Class Hierarchy

C#

```
public class WarehouseApi : CommonApiBase;
```

Visual Basic

```
Public Class WarehouseApi
Inherits CommonApiBase
```

File

WarehouseApi.cs

Description

This is class Ektron.Cms.Commerce.WarehouseApi.

WarehouseApi Methods

The methods of the WarehouseApi class are listed here.

Public Methods

	Name	Description
	Add (see page 668)	This is Add, a member of class WarehouseApi.
	Delete (see page 668)	This is Delete, a member of class WarehouseApi.
	GetDefaultWarehouse (see page 668)	This is GetDefaultWarehouse, a member of class WarehouseApi.
	GetItem (see page 668)	This is GetItem, a member of class WarehouseApi.
	GetList (see page 668)	This is GetList, a member of class WarehouseApi.
	MarkAsDefault (see page 669)	This is MarkAsDefault, a member of class WarehouseApi.
	Update (see page 669)	This is Update, a member of class WarehouseApi.

WarehouseApi.Add Method

C#

```
public void Add(WarehouseData warehouseData);
```

Visual Basic

```
Public Function Add(warehouseData As WarehouseData) As void
```

Description

This is Add, a member of class WarehouseApi.

WarehouseApi.Delete Method

C#

```
public void Delete(long warehouseId);
```

Visual Basic

```
Public Function Delete(warehouseId As long) As void
```

Description

This is Delete, a member of class WarehouseApi.

WarehouseApi.GetDefaultWarehouse Method

C#

```
public WarehouseData GetDefaultWarehouse();
```

Visual Basic

```
Public Function GetDefaultWarehouse() As WarehouseData
```

Description

This is GetDefaultWarehouse, a member of class WarehouseApi.

WarehouseApi.GetItem Method

C#

```
public WarehouseData GetItem(long warehouseId);
```

Visual Basic

```
Public Function GetItem(warehouseId As long) As WarehouseData
```

Description

This is GetItem, a member of class WarehouseApi.

WarehouseApi.GetList Method

C#

```
public List<WarehouseData> GetList(Common.Criteria<WarehouseProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Common.Criteria<WarehouseProperty>) As List<WarehouseData>
```

Description

This is GetList, a member of class WarehouseApi.

WarehouseApi.MarkAsDefault Method

C#

```
public void MarkAsDefault(long warehouseId);
```

Visual Basic

```
Public Function MarkAsDefault(warehouseId As long) As void
```

Description

This is MarkAsDefault, a member of class WarehouseApi.

WarehouseApi.Update Method

C#

```
public void Update(WarehouseData warehouseData);
```

Visual Basic

```
Public Function Update(warehouseData As WarehouseData) As void
```

Description

This is Update, a member of class WarehouseApi.

Ektron.Cms.API Namespace

The API Namespace contains classes for programmatically accessing features typically found in the Workarea.

Namespaces

Name	Description
Analytics (see page 745)	This is namespace Ektron.Cms.API.Analytics.
Community (see page 750)	The Community API contains classes that allow you to manage aspects of your community site.
Content (see page 822)	This is namespace Ektron.Cms.API.Content.
Search (see page 899)	The Search Developer API enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 725), if you chose to install it, has excellent examples of programmatically using the SearchManager (see page 899). It is installed at http://~YourSite~/CMS400Developer/Websearch/WebsearchApi.aspx. Examples are in VB and C#. Here is a list of some of the examples. • Retrieve a document or HTML content that contains the word *cms*. • Retrieve content that contains *research* and has a searchable metadata *wellness* set to Cancer. • Retrieve images that contain *RotatingGraphic1* in their title. • Retrieve geo-content from CMS400.Net.
Task (see page 909)	This is namespace Ektron.Cms.API.Task.
UrlAliasing (see page 924)	This is namespace Ektron.Cms.API.UrlAliasing.
User (see page 950)	This is namespace Ektron.Cms.API.User.

Classes

	Name	Description
	BusinessRule (see page 671)	Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example: • **Condition** - a site visitor visits your site for the first time • **Action** - add a cookie to the site visitor's system and pop up a welcome message With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control... more (see page 671)
	CustomFields (see page 673)	This is class Ektron.Cms.API.CustomFields.
	Folder (see page 683)	This is class Ektron.Cms.API.Folder.
	Library (see page 695)	This is class Ektron.Cms.API.Library.
	Localization (see page 704)	This is class Ektron.Cms.API.Localization.
	Messaging (see page 713)	The Messaging class allows you to work with Private Message Objects.
	Metadata (see page 714)	This is class Ektron.Cms.API.Metadata.
	Permissions (see page 721)	This is class Ektron.Cms.API.Permissions.
	Site (see page 725)	This is class Ektron.Cms.API.Site.
	SiteMap (see page 732)	This is class Ektron.Cms.API.SiteMap.
	Subscription (see page 736)	This is class Ektron.Cms.API.Subscription.
	Utilities (see page 744)	This is class Ektron.Cms.API.Utilities.

Classes

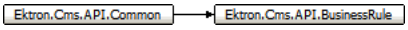
The following table lists classes in this documentation.

Classes

	Name	Description
	BusinessRule (see page 671)	Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example: • **Condition** - a site visitor visits your site for the first time • **Action** - add a cookie to the site visitor's system and pop up a welcome message With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control... more (see page 671)
	CustomFields (see page 673)	This is class Ektron.Cms.API.CustomFields.
	Folder (see page 683)	This is class Ektron.Cms.API.Folder.
	Library (see page 695)	This is class Ektron.Cms.API.Library.
	Localization (see page 704)	This is class Ektron.Cms.API.Localization.
	Messaging (see page 713)	The Messaging class allows you to work with Private Message Objects.
	Metadata (see page 714)	This is class Ektron.Cms.API.Metadata.
	Permissions (see page 721)	This is class Ektron.Cms.API.Permissions.
	Site (see page 725)	This is class Ektron.Cms.API.Site.
	SiteMap (see page 732)	This is class Ektron.Cms.API.SiteMap.
	Subscription (see page 736)	This is class Ektron.Cms.API.Subscription.
	Utilities (see page 744)	This is class Ektron.Cms.API.Utilities.

BusinessRule Class

Class Hierarchy



C#

```
public class BusinessRule : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class BusinessRule
Inherits Ektron.Cms.API.Common
```

File

BusinessRule.vb

Description

Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example:

- **Condition** - a site visitor visits your site for the first time
- **Action** - add a cookie to the site visitor's system and pop up a welcome message

With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control to your Web form, and assign the server control the Business Rules Ruleset.

.NET Assemblies BusinessRule API vs. BusinessRules Server Control API

The .NET Assemblies BusinessRule API (see page 669) allows you to work with results of a Business Rule. The BusinessRules (see page 73) server control API (see page 669) allows you to customize how and where a Business Rule is used on a Web page.

To learn about the BusinessRules (see page 73) server control API (see page 669), click here (see page 73).

BusinessRule Methods

The methods of the BusinessRule class are listed here.

Public Methods

	Name	Description
	GetBusinessRules (see page 671)	This is the overview for the GetBusinessRules method overload.

GetBusinessRules Method

This is the overview for the GetBusinessRules method overload.

Description

Method to get the results from a business rule. This method is overloaded.

Overload List

	Name	Description
	BusinessRule.GetBusinessRules (String, String, String, String, String, String, String, Long) (see page 672)	Method to get the results from a business rule by ID. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset.
	BusinessRule.GetBusinessRules (String, String, String, String, String, String, String, String) (see page 672)	Method to get the results from a business rule by identifier. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset. Identifier is a combination of the path/page/uniqueID of the business rules control.

BusinessRule.GetBusinessRules Method (String, String, String, String, String, String, String, Long)

Method to get the results from a business rule by ID. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset.

C#

```
public String GetBusinessRules(String httpHost, String urlRest, String cookies, String
formValues, String queryValues, String serverValues, String paramValues, Long iRuleID);
```

Visual Basic

```
Public Function GetBusinessRules(ByVal httpHost As String, ByVal urlRest As String, ByVal
cookies As String, ByVal formValues As String, ByVal queryValues As String, ByVal serverValues
As String, ByVal paramValues As String, ByVal iRuleID As Long) As String
```

Parameters

Parameters	Description
ByVal httpHost As String	The HTTP host the request is made on.
ByVal urlRest As String	not used
ByVal cookies As String	Cookie Value Collection (see page 95)
ByVal formValues As String	Form (see page 871) Value Collection (see page 95)
ByVal queryValues As String	Querystring Value Collection (see page 95)
ByVal serverValues As String	Server Variables Collection (see page 95)
ByVal paramValues As String	Custom Value Collection (see page 95)
ByVal iRuleID As Long	Numeric value representing the ID of the Ruleset

Returns

A string representing the output of the evaluated ruleset.

BusinessRule.GetBusinessRules Method (String, String, String, String, String, String, String, String)

Method to get the results from a business rule by identifier. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset. Identifier is a combination of the path/page/uniqueID of the business rules control.

C#

```
public String GetBusinessRules(String httpHost, String urlRest, String cookies, String
formValues, String queryValues, String serverValues, String paramValues, String sRuleID);
```

Visual Basic

```
Public Function GetBusinessRules(ByVal httpHost As String, ByVal urlRest As String, ByVal
cookies As String, ByVal formValues As String, ByVal queryValues As String, ByVal serverValues
As String, ByVal paramValues As String, ByVal sRuleID As String) As String
```

Parameters

Parameters	Description
ByVal httpHost As String	The HTTP host the request is made on.
ByVal urlRest As String	not used
ByVal cookies As String	Cookie Value Collection (see page 95)
ByVal formValues As String	Form (see page 871) Value Collection (see page 95)
ByVal queryValues As String	Querystring Value Collection (see page 95)
ByVal serverValues As String	Server Variables Collection (see page 95)
ByVal paramValues As String	Custom Value Collection (see page 95)
ByVal sRuleID As String	String representing the identifier of the Ruleset. The identifier is a combination of the path/page/uniqueID of the business rules control.

Returns

A string representing the output of the evaluated ruleset.

CustomFields Class

Class Hierarchy

C#

```
public class CustomFields : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class CustomFields
Inherits Ektron.Cms.API.Common
```

File

CustomFields.vb

Description

This is class Ektron.Cms.API.CustomFields.

CustomFields Methods

The methods of the CustomFields class are listed here.

Public Methods

	Name	Description
	AddNewEditableCustomFieldAssignments (see page 674)	Returns the HTML string for the add/edit custom fields.
S	BoolToYesNo (see page 675)	Returns labels for yes or no based on locale.

	GetEditableCustomFieldAssignments (see page 675)	Returns the editable custom fields. Break Inheritance checkbox: • Visible if not at root. • Checked if any fields assigned without inheritance. • If unchecked & form saved, removes all assignments. • If checked & form saved and no assignments chosen, creates NULL assignment (CustomFieldID -1). • When checked, enables all assignment checkboxes (iterate through all of this tables elements). • When unchecked, disables all assignment checkboxes (iterate through all of this tables elements). • Assignment Checkboxes: • When clicked: if 'Assignment' button is going unchecked, then ensure required is unchecked. • When clicked: if 'Required' button, don't accept checks if assigned is unchecked. • When form submitted, copy checked... more (see page 675)
S	GetEnhancedMetadataArea (see page 676)	Returns the HTML needed to render the EnhancedMetadata selection area.
S	GetEnhancedMetadataScript (see page 676)	Returns support Javascript code for enhanced metadata.
	GetFieldsByFolder (see page 676)	Returns all of the custom attributes by folder.
S	GetSearchProperties (see page 677)	This is the overview for the GetSearchProperties method overload.
S	Populate_AssetRequestObjectFromForm (see page 678)	Returns asset request object using form data.
	ProcessCustomFields (see page 679)	Process Folders' Custom-Field (and metadata) assignments: When form submitted, copy checked items to hidden field. Form (see page 871) processor: • If break inheritance checked, and any items in hidden field, send contents to db. • If break inheritance checked, and hidden field empty, create dummy assignment fieldID = -1. • If break inheritance NOT checked, remove all assignments for this folder.
S	SearchAllAssets (see page 679)	Returns the search result based on assets.
S	WriteFilteredMetadataForEdit (see page 679)	Returns an HTML string for filtered metadata.
S	WriteFilteredMetadataForView (see page 680)	Returns an HTML string for metadata view.
S	WriteLibrarySearchExtended (see page 680)	Returns an HTML string for the library search custom metadata properties.
S	WriteMetadataDefaultForEdit (see page 681)	Returns an HTML string for the default metadata.
S	WriteMetadataTypeForView (see page 681)	Returns a metadata type for view.
S	WritePublicSearch (see page 681)	Returns WritepublicsearchEx.
S	WritePublicSearchEx (see page 682)	Returns an HTML string for extended the search attributes.

CustomFields.AddNewEditableCustomFieldAssignments Method

Returns the HTML string for the add/edit custom fields.

C#

```
public String AddNewEditableCustomFieldAssignments(Long parentfolderid);
```

Visual Basic

```
Public Function AddNewEditableCustomFieldAssignments(ByVal parentfolderid As Long) As String
```

Parameters

Parameters	Description
ByVal parentfolderid As Long	The parent folder's ID.

Returns

String

CustomFields.BoolToYesNo Method

Returns labels for yes or no based on locale.

C#

```
public Shared String BoolToYesNo(Boolean bValue);
```

Visual Basic

```
Public Shared Function BoolToYesNo(ByVal bValue As Boolean) As String
```

Parameters

Parameters	Description
ByVal bValue As Boolean	True or False

Returns

String

CustomFields.GetEditableCustomFieldAssignments Method

Returns the editable custom fields.

Break Inheritance checkbox:

- Visible if not at root.
- Checked if any fields assigned without inheritance.
- If unchecked & form saved, removes all assignments.
- If checked & form saved and no assignments chosen, creates NULL assignment (CustomFieldID -1).
- When checked, enables all assignment checkboxes (iterate through all of this tables elements).
- When unchecked, disables all assignment checkboxes (iterate through all of this tables elements).
- Assignment Checkboxes:
- When clicked: if 'Assignment' button is going unchecked, then ensure required is unchecked.
- When clicked: if 'Required' button, don't accept checks if assigned is unchecked.
- When form submitted, copy checked items to hidden field.
- Form (see page 871) processor;
- If break inheritance checked, and any items in hidden field, send contents to db.
- If break inheritance checked, and hidden field empty, create dummy assignment fieldID = -1.
- If break inheritance NOT checked, remove all assignments for this folder.

Note: If editMode is false, then items are visible but not changeable by user.

C#

```
public String GetEditableCustomFieldAssignments(Object folderID, Object editMode);
```

Visual Basic

```
Public Function GetEditableCustomFieldAssignments(ByVal folderID As Object, ByVal editMode As
Object) As String
```

Parameters

Parameters	Description
ByVal folderID As Object	The folder's ID.
ByVal editMode As Object	Edit mode.

Returns

String

Remarks

Function GetEditableCustomFieldAssignments:

CustomFields.GetEnhancedMetadataArea Method

Returns the HTML needed to render the EnhancedMetadata selection area.

C#

```
public Shared String GetEnhancedMetadataArea();
```

Visual Basic

```
Public Shared Function GetEnhancedMetadataArea() As String
```

CustomFields.GetEnhancedMetadataScript Method

Returns support Javascript code for enhanced metadata.

C#

```
public Shared String GetEnhancedMetadataScript();
```

Visual Basic

```
Public Shared Function GetEnhancedMetadataScript() As String
```

Returns

String

CustomFields.GetFieldsByFolder Method

Returns all of the custom attributes by folder.

C#

```
public Collection GetFieldsByFolder(Long FolderID, int LanguageID);
```

Visual Basic

```
Public Function GetFieldsByFolder(ByVal FolderID As Long, ByVal LanguageID As Integer) As
Collection
```

Parameters

Parameters	Description
ByVal FolderID As Long	The folder's ID.
ByVal LanguageID As Integer	The language's ID.

Returns

Microsoft.VisualBasic.Collection (see page 95)

GetSearchProperties Method

This is the overview for the GetSearchProperties method overload.

Description

Returns the search's properties. This method is overloaded.

Overload List

	Name	Description
	CustomFields.GetSearchProperties (Ektron.Cms.Common.SearchAssetDispayRequest) (see page 677)	Returns an HTML string for the search properties display.
	CustomFields.GetSearchProperties (String, String, String, String, Collection, String, String, String) (see page 677)	Returns an HTML string to display search properties.
	CustomFields.GetSearchProperties (System.Collections.Specialized.NameValueCollection, Ektron.Cms.Common.SearchItemCustom, Ektron.Cms.Common.SearchItemCustom) (see page 678)	Returns the search's properties.

CustomFields.GetSearchProperties Method (Ektron.Cms.Common.SearchAssetDispayRequest)

Returns an HTML string for the search properties display.

C#

```
public Shared System.Text.StringBuilder
GetSearchProperties(Ektron.Cms.Common.SearchAssetDispayRequest sadReq);
```

Visual Basic

```
Public Shared Function GetSearchProperties(ByVal sadReq As
Ektron.Cms.Common.SearchAssetDispayRequest) As System.Text.StringBuilder
```

Parameters

Parameters	Description
ByVal sadReq As Ektron.Cms.Common.SearchAssetDispayRequest	The search asset objects. (Required)

Returns

System.Text.StringBuilder

CustomFields.GetSearchProperties Method (String, String, String, String, Collection, String, String, String)

Returns an HTML string to display search properties.

C#

```
public Shared Object GetSearchProperties(String MonikerStyle, String DataType, String Caption,
String Name, Collection Items, String AlignCaption, String szBeginFontInfo, String
szEndFontInfo);
```

Visual Basic

```
Public Shared Function GetSearchProperties(ByVal MonikerStyle As String, ByVal DataType As
```

```
String, ByVal Caption As String, ByVal Name As String, ByVal Items As Collection, ByVal
AlignCaption As String, ByVal szBeginFontInfo As String, ByVal szEndFontInfo As String) As
Object
```

Parameters

Parameters	Description
ByVal MonikerStyle As String	Style. (Required)
ByVal DataType As String	The data type such as String, DateTime, Numeric, etc.
ByVal Caption As String	The caption value. (Required)
ByVal Name As String	The value for name attributes.
ByVal Items As Collection	Values of items.
ByVal AlignCaption As String	The value for the align attributes.
ByVal szBeginFontInfo As String	The value for the begin font attributes.
ByVal szEndFontInfo As String	The value for the end font attributes.

Returns

Object

CustomFields.GetSearchProperties Method (System.Collections.Specialized.NameValueCollection, Ektron.Cms.Common.SearchItemCustom, Ektron.Cms.Common.SearchItemCustom)

Returns the search's properties.

C#

```
public Shared GetSearchProperties(System.Collections.Specialized.NameValueCollection FormReq,
ref Ektron.Cms.Common.SearchItemCustom CustomProps, ref Ektron.Cms.Common.SearchItemCustom
AssetProps);
```

Visual Basic

```
Public Shared Sub GetSearchProperties(ByVal FormReq As
System.Collections.Specialized.NameValueCollection, ByRef CustomProps() As
Ektron.Cms.Common.SearchItemCustom, ByRef AssetProps() As Ektron.Cms.Common.SearchItemCustom)
```

Parameters

Parameters	Description
ByVal FormReq As System.Collections.Specialized.NameValueCollection	Form (see page 871) field values.
ByRef CustomProps() As Ektron.Cms.Common.SearchItemCustom	Custom properties value.
ByRef AssetProps() As Ektron.Cms.Common.SearchItemCustom	Asset (see page 822) properties value.

CustomFields.Populate_AssetRequestObjectFromForm Method

Returns asset request object using form data.

C#

```
public Shared Ektron.Cms.Common.SearchAssetRequest
Populate_AssetRequestObjectFromForm(Optional ByVal
System.Collections.Specialized.NameValueCollection PostBackData);
```

Visual Basic

```
Public Shared Function Populate_AssetRequestObjectFromForm(Optional ByVal PostBackData As
System.Collections.Specialized.NameValueCollection = Nothing) As
Ektron.Cms.Common.SearchAssetRequest
```

Returns

Ektron.Cms.Common.SearchAssetRequest

CustomFields.ProcessCustomFields Method

Process Folders' Custom-Field (and metadata) assignments:

When form submitted, copy checked items to hidden field.

Form (see page 871) processor:

- If break inheritance checked, and any items in hidden field, send contents to db.
- If break inheritance checked, and hidden field empty, create dummy assignment fieldID = -1.
- If break inheritance NOT checked, remove all assignments for this folder.

C#

```
public Object ProcessCustomFields(Long folderID);
```

Visual Basic

```
Public Function ProcessCustomFields(ByVal folderID As Long) As Object
```

Parameters

Parameters	Description
ByVal folderID As Long	The folder's ID.

Returns

Object

Remarks

CustomFields.SearchAllAssets Method

Returns the search result based on assets.

C#

```
public Shared System.Text.StringBuilder SearchAllAssets(SearchAssetDispayRequest sadReq);
```

Visual Basic

```
Public Shared Function SearchAllAssets(ByVal sadReq As SearchAssetDispayRequest) As
System.Text.StringBuilder
```

Parameters

Parameters	Description
ByVal sadReq As SearchAssetDispayRequest	Search (see page 899) parameters. (Required)

Returns

System.Text.StringBuilder

CustomFields.WriteFilteredMetadataForEdit Method

Returns an HTML string for filtered metadata.

C#

```
public Shared System.Text.StringBuilder
WriteFilteredMetadataForEdit(Ektron.Cms.ContentMetaData cMetadataTypes, Boolean
bSearchableOnly, String ty, Long FolderId, Optional ByRef int Count, Optional ByVal Hashtable
metaPermsObj);
```

Visual Basic

```
Public Shared Function WriteFilteredMetadataForEdit(ByVal cMetadataTypes() As
Ektron.Cms.ContentMetaData, ByVal bSearchableOnly As Boolean, ByVal ty As String, ByVal
FolderId As Long, Optional ByRef Count As Integer = 0, Optional ByVal metaPermsObj As
Hashtable = Nothing) As System.Text.StringBuilder
```

Parameters

Parameters	Description
ByVal cMetadataTypes() As Ektron.Cms.ContentMetaData	Metadata (see page 714) types.
ByVal bSearchableOnly As Boolean	bSearchableOnly
ByVal ty As String	type
ByVal FolderId As Long	The folder's ID.
Optional ByRef Count As Integer = 0	Number of items. (Required)

Returns

System.Text.StringBuilder

CustomFields.WriteFilteredMetadataForView Method

Returns an HTML string for metadata view.

C#

```
public Shared String WriteFilteredMetadataForView(Ektron.Cms.ContentMetaData cMetadataTypes,
Long FolderId, Boolean bSearchableOnly);
```

Visual Basic

```
Public Shared Function WriteFilteredMetadataForView(ByVal cMetadataTypes() As
Ektron.Cms.ContentMetaData, ByVal FolderId As Long, ByVal bSearchableOnly As Boolean) As String
```

Parameters

Parameters	Description
ByVal cMetadataTypes() As Ektron.Cms.ContentMetaData	Metadata (see page 714) types.
ByVal FolderId As Long	The folder's Id.
ByVal bSearchableOnly As Boolean	True or False

Returns

String

CustomFields.WriteLibrarySearchExtended Method

Returns an HTML string for the library search custom metadata properties.

C#

```
public Shared System.Text.StringBuilder WriteLibrarySearchExtended(Long folderId);
```

Visual Basic

```
Public Shared Function WriteLibrarySearchExtended(ByVal folderId As Long) As
System.Text.StringBuilder
```

Parameters

Parameters	Description
ByVal folderId As Long	The folder's ID.

Returns

System.Text.StringBuilder

CustomFields.WriteMetadataDefaultForEdit Method

Returns an HTML string for the default metadata.

C#

```
public Shared System.Text.StringBuilder WriteMetadataDefaultForEdit(Ektron.Cms.ContentMetaData
cMetadataType, int DataType, Optional ByVal Hashtable metaPermsObj);
```

Visual Basic

```
Public Shared Function WriteMetadataDefaultForEdit(ByVal cMetadataType As
Ektron.Cms.ContentMetaData, ByVal DataType As Integer, Optional ByVal metaPermsObj As
Hashtable = Nothing) As System.Text.StringBuilder
```

Parameters

Parameters	Description
ByVal cMetadataType As Ektron.Cms.ContentMetaData	Metadata (see page 714) values.
ByVal DataType As Integer	Data types

Returns

System.Text.StringBuilder

CustomFields.WriteMetadataTypeForView Method

Returns a metadata type for view.

C#

```
public Shared String WriteMetadataTypeForView(String DataType, String Caption, String Value);
```

Visual Basic

```
Public Shared Function WriteMetadataTypeForView(ByVal DataType As String, ByVal Caption As
String, ByVal Value As String) As String
```

Parameters

Parameters	Description
ByVal DataType As String	Data types.
ByVal Caption As String	Caption value.
ByVal Value As String	Value. (Required)

Returns

String

CustomFields.WritePublicSearch Method

Returns WritepublicsearchEx.

C#

```
public Shared System.Text.StringBuilder WritePublicSearch(SearchAssetDispayRequest sadReq);
```

Visual Basic

```
Public Shared Function WritePublicSearch(ByVal sadReq As SearchAssetDispayRequest) As
System.Text.StringBuilder
```

Parameters

Parameters	Description
ByVal sadReq As SearchAssetDispayRequest	Search (see page 899) asset data. (Required)

Returns

System.Text.StringBuilder

CustomFields.WritePublicSearchEx Method

Returns an HTML string for extended the search attributes.

C#

```
public Shared System.Text.StringBuilder WritePublicSearchEx(SearchAssetDispayRequest sadReq,
Boolean EnableAdvancedLink, Boolean EnableContentBtn, Boolean EnableContent, Boolean
EnableFormsBtn, Boolean EnableForms, Boolean EnableDMSAssetsBtn, Boolean EnableDMSAssets,
Boolean EnableLibImagesBtn, Boolean EnableLibImages, Boolean EnableLibFilesBtn, Boolean
EnableLibFiles, Boolean EnableLibHyperlinksBtn, Boolean EnableLibHyperlinks, Boolean
EnableArchivedBtn, Boolean EnableArchived, String TargetFolderDescriptionList, Optional ByVal
Boolean EnableContentFieldset, Optional ByVal String ContentFieldsetLegend, Optional ByVal
Boolean EnableLibraryFieldset, Optional ByVal String LibraryFieldsetLegend, Optional ByVal
Boolean EnableFolderFieldset, Optional ByVal String FolderFieldsetLegend, Optional ByVal
Boolean EnableCallBack, Optional ByVal Boolean RemoveTeaserHtml, Optional ByVal int
MaxTeaserLength);
```

Visual Basic

```
Public Shared Function WritePublicSearchEx(ByVal sadReq As SearchAssetDispayRequest, ByVal
EnableAdvancedLink As Boolean, ByVal EnableContentBtn As Boolean, ByVal EnableContent As
Boolean, ByVal EnableFormsBtn As Boolean, ByVal EnableForms As Boolean, ByVal
EnableDMSAssetsBtn As Boolean, ByVal EnableDMSAssets As Boolean, ByVal EnableLibImagesBtn As
Boolean, ByVal EnableLibImages As Boolean, ByVal EnableLibFilesBtn As Boolean, ByVal
EnableLibFiles As Boolean, ByVal EnableLibHyperlinksBtn As Boolean, ByVal EnableLibHyperlinks
As Boolean, ByVal EnableArchivedBtn As Boolean, ByVal EnableArchived As Boolean, ByVal
TargetFolderDescriptionList As String, Optional ByVal EnableContentFieldset As Boolean =
False, Optional ByVal ContentFieldsetLegend As String = "", Optional ByVal
EnableLibraryFieldset As Boolean = False, Optional ByVal LibraryFieldsetLegend As String = "",
Optional ByVal EnableFolderFieldset As Boolean = False, Optional ByVal FolderFieldsetLegend As
String = "", Optional ByVal EnableCallBack As Boolean = False, Optional ByVal RemoveTeaserHtml
As Boolean = True, Optional ByVal MaxTeaserLength As Integer = 100) As
System.Text.StringBuilder
```

Parameters

Parameters	Description
ByVal sadReq As SearchAssetDispayRequest	Search (see page 899) asset properties. (Required)
ByVal EnableAdvancedLink As Boolean	True or False
ByVal EnableContentBtn As Boolean	True or False
ByVal EnableContent As Boolean	True or False
ByVal EnableFormsBtn As Boolean	True or False
ByVal EnableForms As Boolean	True or False
ByVal EnableDMSAssetsBtn As Boolean	True or False
ByVal EnableDMSAssets As Boolean	True or False
ByVal EnableLibImagesBtn As Boolean	True or False
ByVal EnableLibImages As Boolean	True or False
ByVal EnableLibFilesBtn As Boolean	True or False
ByVal EnableLibFiles As Boolean	True or False
ByVal EnableLibHyperlinksBtn As Boolean	True or False
ByVal EnableLibHyperlinks As Boolean	True or False
ByVal EnableArchivedBtn As Boolean	True or False
ByVal EnableArchived As Boolean	True or False
ByVal TargetFolderDescriptionList As String	True or False
Optional ByVal EnableContentFieldset As Boolean = False	True or False
Optional ByVal ContentFieldsetLegend As String = ""	True or False

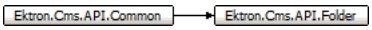
Optional ByVal EnableLibraryFieldset As Boolean = False	True or False
Optional ByVal LibraryFieldsetLegend As String = ""	True or False
Optional ByVal EnableFolderFieldset As Boolean = False	True or False
Optional ByVal FolderFieldsetLegend As String = ""	True or False

Returns

System.Text.StringBuilder

Folder Class

Class Hierarchy



C#

```
public class Folder : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Folder
Inherits Ektron.Cms.API.Common
```

File

Folder.vb

Description

This is class Ektron.Cms.API.Folder.

Folder Methods

The methods of the Folder class are listed here.

Public Methods

	Name	Description
	AddCommunityFolder (see page 684)	Creates CMS Community (see page 750) folder.
	AddFolder (see page 684)	Creates a new folder in the CMS. By default the new folder inherits subscription properties of parent folder. To override, use the BreakInheritance property of the FolderRequest.SubscriptionProperties object (set it to True). *See Also:* The Example section.
V	CreateFolder (see page 685)	Creates a folder.
	DeleteFolderById (see page 685)	Delete the folder by its ID. *See Also:* The Example section.
	GetChildFolders (see page 687)	This is the overview for the GetChildFolders method overload.
	GetFolder (see page 690)	This is the overview for the GetFolder method overload.
	GetFolderDataWithPermission (see page 691)	Retrieves the folder's details with permission information.
	GetFolderId (see page 692)	This is the overview for the GetFolderId method overload.
V	GetFolderInfo (see page 693)	Returns the folder detail.
	GetPath (see page 693)	Gets the path of a folder in the Workarea by using the folder's ID.
	IsFolderHidden (see page 693)	Returns flag indicating if the folder is hidden.
	RenameFolder (see page 694)	Rename the folder with new name.

	UpdateFolder (see page 694)	This is the overview for the UpdateFolder method overload.
	UpdateFolderApprovalMethod (see page 695)	Updates the ApprovalMethod of a folder. The ApprovalMethod gets or set whether or not to force all approvers to approve the content that is added to the folder. Before using this method, you must break permission inheritance for the folder by setting the FolderRequest (see page 1615) object's BreakInheritButton property to True.

Folder.AddCommunityFolder Method

Creates CMS Community (see page 750) folder.

C#

```
public AddCommunityFolder(FolderRequest request);
```

Visual Basic

```
Public Sub AddCommunityFolder(ByVal request As FolderRequest)
```

Parameters

Parameters	Description
ByVal request As FolderRequest	The Folder (see page 683)'s properties as a FolderRequest (see page 1615) object.

Folder.AddFolder Method

Creates a new folder in the CMS. By default the new folder inherits subscription properties of parent folder. To override, use the BreakInheritance property of the FolderRequest.SubscriptionProperties object (set it to True).

See Also: The Example section.

C#

```
public AddFolder(FolderRequest request);
```

Visual Basic

```
Public Sub AddFolder(ByVal request As FolderRequest)
```

Parameters

Parameters	Description
ByVal request As FolderRequest	The Folder (see page 683)'s properties as a FolderRequest (see page 1615) object.

Example

The following example shows how to create a Web page from which a folder can be added to CMS400.NET. This example uses some standard drag and drop controls and a small section of VB code utilizing the AddFolder method. This method uses the InternalAdmin to add the folder. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label:
```
<asp:Label ID="Label1" runat="server" Text="Folder Name: " Width="84px"></asp:Label>
```
2. Add a text box to enter the folder's name.
```
<asp:TextBox ID="txtFolderName" runat="server" Width="96px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder Name box.
```
<asp:RequiredFieldValidator ID="rfvFolderName" runat="server"
ControlToValidate="txtFolderName"
  ErrorMessage="Please enter folder name"></asp:RequiredFieldValidator>
```
4. Add a couple of breaks for spacing purposes.
```
<br /><br />
```
5. Add a label for the description box.
```
<asp:Label ID="Label2" runat="server" Text="Description: " Width="72px"></asp:Label>
```
6. Add a text box to enter a description.
```
<asp:TextBox ID="txtDescription" runat="server" Width="358px"></asp:TextBox>
```
7. Add two more breaks for spacing purposes.

```
<br /><br />
```

8. Add a button to submit the information. We will set the button click event in the code behind.
```
<asp:Button ID="btnAddFolder" runat="server" Text="Add CMS Folder" Width="110px" />
```
9. Add two last breaks.
```
<br /><br />
```
10. Add a results label that lets a user know if the add succeeded.
```
<asp:Label ID="lblResult" runat="server" Width="545px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.
```
        Protected Sub btnAddFolder_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnAddFolder.Click
```
2. Create an object for the Folder (see page 683) API (see page 669).
```
                Dim folderApi As New Ektron.Cms.API.Folder
```
3. Create a Try/Catch around the AddFolder method. This allows you to catch any exceptions and helps when debugging.
4. Create a FolderRequest (see page 1615) object and assign the following properties:
```
                Try
                    Dim req As New Ektron.Cms.FolderRequest
                    req.FolderName = txtFolderName.Text
                    req.FolderDescription = txtDescription.Text
                    req.ParentId = 0  'Add this folder under the root folder in the Workarea
                    req.StyleSheet = ""
                    folderApi.AddFolder(req)
```
5. Display the folder name and ID and the fact that it was added.
```
                    Dim newFolderID As Integer = req.FolderId
                    lblResult.Text = "Folder " & txtFolderName.Text & " was added under the
root folder in the Workarea with FolderID = " & newFolderId.ToString()
```
6. If there is an error, let the user know what happened.
```
                Catch ex As Exception
                    lblResult.Text = ex.Message
                End Try

        End Sub
```

Folder.CreateFolder Method

Creates a folder.

C#

```
public Overridable Boolean CreateFolder(String Path);
```

Visual Basic

```
Public Overridable Function CreateFolder(ByVal Path As String) As Boolean
```

Parameters

Parameters	Description
ByVal Path As String	The path on the IIS server. (Required)

Returns

Boolean

Folder.DeleteFolderById Method

Delete the folder by its ID.

See Also: The Example section.

C#

```
public Boolean DeleteFolderById(Long Id);
```

Visual Basic

```
Public Function DeleteFolderById(ByVal Id As Long) As Boolean
```

Parameters

Parameters	Description
ByVal Id As Long	The folder's ID.

Returns

False = successfully deleted. Otherwise, returns True.

Example

The following example shows how to create a Web page from which a folder can be deleted from CMS400.NET by entering its ID. This example uses some standard drag and drop controls and a small section of VB code utilizing the DeleteFolderById method. This method uses the InternalAdmin to delete the folder. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label:
```
<asp:Label ID="Label1" runat="server" Text="Folder ID: "></asp:Label>
```
2. Add a text box to enter the folder's ID.
```
<asp:TextBox ID="txtFolderId" runat="server" Width="44px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder ID box.
```
<asp:RequiredFieldValidator ID="rfvFolderId" runat="server" ControlToValidate="txtFolderId"
  ErrorMessage="Please enter folder ID"></asp:RequiredFieldValidator>
```
4. Add a couple of breaks for spacing purposes.
```
<br /><br />
```
5. Add a delete button. We will set the button click event in the code behind.
```
<asp:Button ID="btnDeleteFolder" runat="server" Text="Delete Folder" Width="95px" />
```
6. Add two more breaks for spacing purposes.
```
<br /><br />
```
7. Add a label to let the user know if the delete succeeded.
```
<asp:Label ID="lblResult" runat="server" Width="346px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.
```
        Protected Sub btnDeleteFolder_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnDeleteFolder.Click
```
2. Check to make sure the folder ID entered is numeric.
```
            If Not IsNumeric(txtFolderId.Text) Then
                lblResult.Text = "Invalid Folder ID"
                Exit Sub
            End If
```
3. Create an object for the Folder (see page 683) API (see page 669).
```
            Dim folderApi As New Ektron.Cms.API.Folder
```
4. Create a Try/Catch around the DeleteFolderById method. This allows you to catch any exceptions and helps when debugging.
```
            Try
                folderApi.DeleteFolderById(CInt(txtFolderId.Text))
```
5. Display whether the deletion was successful.
```
                lblResult.Text = "The above folder was deleted from CMS400.NET"
```
6. If there is an error, let the user know.

```
            Catch ex As Exception
                lblResult.Text = "Error deleting folder"
            End Try

    End Sub
```

GetChildFolders Method

This is the overview for the GetChildFolders method overload.

Overload List

	Name	Description
	Folder.GetChildFolders (Long, Boolean, EkEnumeration.FolderOrderBy) (see page 687)	Loads all the subfolders for a given folder ID. *See Also:* The Example section.
	Folder.GetChildFolders (Long, Boolean, String) (see page 688)	Loads all the subfolders for a given folder ID. *See Also:* The Example section.

Folder.GetChildFolders Method (Long, Boolean, EkEnumeration.FolderOrderBy)

Loads all the subfolders for a given folder ID.

See Also: The Example section.

C#

```
public FolderData[] GetChildFolders(Long id, Boolean recursive, EkEnumeration.FolderOrderBy
orderBy);
```

Visual Basic

```
Public Function GetChildFolders(ByVal id As Long, ByVal recursive As Boolean, ByVal orderBy As
EkEnumeration.FolderOrderBy) As FolderData()
```

Parameters

Parameters	Description
ByVal id As Long	The folder's ID
ByVal recursive As Boolean	True = recursive look up. False = none.
ByVal orderBy As EkEnumeration.FolderOrderBy	Field to order results by.

Returns

Ektron.Cms.FolderData()

Example

The following example shows how to create a Web page from which a folder ID can be entered and list of child folders returned. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetChildFolders method. This method uses the InternalAdmin to retrieve the child folder list. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label:
```
<asp:Label ID="Label1" runat="server" Text="Folder ID: "></asp:Label>
```
2. Add a text box to enter the folder's ID.
```
<asp:TextBox ID="txtFolderId" runat="server" Width="44px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder ID box.
```
<asp:RequiredFieldValidator ID="rfvFolderId" runat="server" ControlToValidate="txtFolderId"
  ErrorMessage="Please enter folder ID"></asp:RequiredFieldValidator>
```

4. Add a couple of breaks for spacing purposes.
```
<br /><br />
```
5. Add a submit button. We will set the button click event in the code behind.
```
<asp:Button ID="btnGetChildFolders" runat="server" Text="Get Child Folders" Width="123px" />
```
6. Add two more breaks for spacing purposes.
```
<br /><br />
```
7. Add a label to display the list of child folder names.
```
<asp:Label ID="lblResult" runat="server" Width="272px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.
```
        Protected Sub btnGetChildFolders_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnGetChildFolders.Click
```
2. Create an object for the Folder (see page 683) API (see page 669).
```
            Dim folderApi As New Ektron.Cms.API.Folder
```
3. Check to make sure the folder ID is numeric.
```
            If Not IsNumeric(txtFolderId.Text) Then
                lblResult.Text = "Invalid Folder ID"
                Exit Sub
            End If
```
4. Create an object as an array of FolderData.
```
            Dim folders As Ektron.Cms.FolderData()
```
5. Create a Try/Catch around the GetChildFolders method. This allows you to catch any exceptions and helps when debugging.
```
            Try
                folders = folderApi.GetChildFolders(CInt(txtFolderId.Text), True,
FolderOrderBy.Name)
```
6. Use an If/Else statement to display whether child folders exist.
```
                If (folders Is Nothing) Then
                    lblResult.Text &= "Child folders: None"
```
7. If child folders exist, display a total count.
```
                Else
                    lblResult.Text &= "Child folders: " & folders.Length.ToString & "<br/>"
```
8. Iterate through the array and display each child folder's name.
```
                    Dim i As Integer
                    For i = 0 To folders.Length - 1 Step 1
                        lblResult.Text &= folders(i).Name & "<br/>"
                    Next
                End If
```
9. Display a message if there is an error.
```
            Catch ex As Exception
                lblResult.Text = "Error getting child folders"
            End Try

        End Sub
```
10. Add a page load event so that lblResult.Text is blank when the page first loads.
```
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        lblResult.Text = ""
    End Sub
```

Folder.GetChildFolders Method (Long, Boolean, String)

Loads all the subfolders for a given folder ID.

See Also: The Example section.

C#

```
[Obsolete("Use new GetChildFolders with FolderOrderBy parameter.")]
public FolderData[] GetChildFolders(Long Id, Optional ByVal Boolean Recursive, Optional ByVal
String OrderBy);
```

Visual Basic

```
<Obsolete("Use new GetChildFolders with FolderOrderBy parameter.")>
Public Function GetChildFolders(ByVal Id As Long, Optional ByVal Recursive As Boolean = False,
Optional ByVal OrderBy As String = "name") As FolderData()
```

Parameters

Parameters	Description
ByVal Id As Long	The folder's ID
Optional ByVal Recursive As Boolean = False	True = recursive look up. False = none.
Optional ByVal OrderBy As String = "name"	An order by clause

Returns

Ektron.Cms.FolderData()

Example

The following example shows how to create a Web page from which a folder ID can be entered and list of child folders returned. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetChildFolders method. This method uses the InternalAdmin to retrieve the child folder list. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label:
```
<asp:Label ID="Label1" runat="server" Text="Folder ID: "></asp:Label>
```
2. Add a text box to enter the folder's ID.
```
<asp:TextBox ID="txtFolderId" runat="server" Width="44px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder ID box.
```
<asp:RequiredFieldValidator ID="rfvFolderId" runat="server" ControlToValidate="txtFolderId"
  ErrorMessage="Please enter folder ID"></asp:RequiredFieldValidator>
```
4. Add a couple of breaks for spacing purposes.
```
<br /><br />
```
5. Add a submit button. We will set the button click event in the code behind.
```
<asp:Button ID="btnGetChildFolders" runat="server" Text="Get Child Folders" Width="123px" />
```
6. Add two more breaks for spacing purposes.
```
<br /><br />
```
7. Add a label to display the list of child folder names.
```
<asp:Label ID="lblResult" runat="server" Width="272px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.
```
Protected Sub btnGetChildFolders_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnGetChildFolders.Click
```
2. Create an object for the Folder (see page 683) API (see page 669).
```
Dim folderApi As New Ektron.Cms.API.Folder
```
3. Check to make sure the folder ID is numeric.
```
If Not IsNumeric(txtFolderId.Text) Then
    lblResult.Text = "Invalid Folder ID"
    Exit Sub
End If
```
4. Create an object as an array of FolderData.

```
Dim folders As Ektron.Cms.FolderData()
```

5. Create a Try/Catch around the GetChildFolders method. This allows you to catch any exceptions and helps when debugging.

```
Try
    folders = folderApi.GetChildFolders(CInt(txtFolderId.Text), True, "name")
```

6. Use an If/Else statement to display whether child folders exist.

```
    If (folders Is Nothing) Then
        lblResult.Text &= "Child folders: None"
```

7. If child folders exist, display a total count.

```
    Else
        lblResult.Text &= "Child folders: " & folders.Length.ToString & "<br/>"
```

8. Iterate through the array and display each child folder's name.

```
        Dim i As Integer
        For i = 0 To folders.Length - 1 Step 1
            lblResult.Text &= folders(i).Name & "<br/>"
        Next
    End If
```

9. Display a message if there is an error.

```
Catch ex As Exception
   lblResult.Text = "Error getting child folders"
End Try

End Sub
```

10. Add a page load event so that lblResult.Text is blank when the page first loads.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
   lblResult.Text = ""
End Sub
```

GetFolder Method

This is the overview for the GetFolder method overload.

Overload List

	Name	Description
	Folder.GetFolder (Long) (see page 690)	Retrieves the folder's details.
	Folder.GetFolder (Long, Boolean) (see page 691)	Retrieves the folder's details with taxonomy information.
	Folder.GetFolder (Long, Boolean, Boolean) (see page 691)	Retrieves the folder's details with taxonomy information and Flag (see page 1231) information

Folder.GetFolder Method (Long)

Retrieves the folder's details.

C#

```
public FolderData GetFolder(Long Id);
```

Visual Basic

```
Public Function GetFolder(ByVal Id As Long) As FolderData
```

Parameters

Parameters	Description
ByVal Id As Long	The Folder (see page 683)'s ID.

Returns

Ektron.Cms.FolderData

Remarks

Read permission required

Folder.GetFolder Method (Long, Boolean)

Retrieves the folder's details with taxonomy information.

C#

```
public FolderData GetFolder(Long Id, Boolean GetTaxonomy);
```

Visual Basic

```
Public Function GetFolder(ByVal Id As Long, ByVal GetTaxonomy As Boolean) As FolderData
```

Parameters

Parameters	Description
ByVal Id As Long	The Folder (see page 683)'s ID.
ByVal GetTaxonomy As Boolean	If true, gets the Taxonomy (see page 876) associated with the folder information.

Returns

Ektron.Cms.FolderData

Folder.GetFolder Method (Long, Boolean, Boolean)

Retrieves the folder's details with taxonomy information and Flag (see page 1231) information

C#

```
public FolderData GetFolder(Long Id, Boolean GetTaxonomy, Boolean GetFlags);
```

Visual Basic

```
Public Function GetFolder(ByVal Id As Long, ByVal GetTaxonomy As Boolean, ByVal GetFlags As
Boolean) As FolderData
```

Parameters

Parameters	Description
ByVal Id As Long	The Folder (see page 683)'s ID.
ByVal GetTaxonomy As Boolean	If true, gets the Taxonomy (see page 876) associated with the folder information.
ByVal GetFlags As Boolean	If true, gets the Flag (see page 1231) definitions associated with the folder

Returns

Ektron.Cms.FolderData

Folder.GetFolderDataWithPermission Method

Retrieves the folder's details with permission information.

C#

```
public FolderData GetFolderDataWithPermission(Long Id);
```

Visual Basic

```
Public Function GetFolderDataWithPermission(ByVal Id As Long) As FolderData
```

Parameters

Parameters	Description
ByVal Id As Long	The folder's numeric ID.

Returns

Ektron.Cms.FolderData

GetFolderId Method

This is the overview for the GetFolderId method overload.

Description

This method is overloaded. Finds the folder ID by using the content's ID. In addition, you could choose to find the folder ID by using the folder's name and naming the parent folder's ID.

Overload List

	Name	Description
	Folder.GetFolderId (Long) (see page 692)	Finds the folder ID by using a content's ID.
	Folder.GetFolderId (String, Long) (see page 692)	Finds the folder ID by specifying its folder name and the parent folder ID.

Folder.GetFolderId Method (Long)

Finds the folder ID by using a content's ID.

C#

```
public Long GetFolderId(Long Id);
```

Visual Basic

```
Public Function GetFolderId(ByVal Id As Long) As Long
```

Parameters

Parameters	Description
ByVal Id As Long	The content's numeric ID.

Returns

A numeric value representing the folder ID.

Folder.GetFolderId Method (String, Long)

Finds the folder ID by specifying its folder name and the parent folder ID.

C#

```
public Long GetFolderId(String FolderName, Long ParentId);
```

Visual Basic

```
Public Function GetFolderId(ByVal FolderName As String, ByVal ParentId As Long) As Long
```

Parameters

Parameters	Description
ByVal FolderName As String	Name of the folder.
ByVal ParentId As Long	The numeric ID of the parent folder.

Returns

Returns the folder ID. Returns 0 if it does not exist.

Folder.GetFolderInfo Method

Returns the folder detail.

C#

```
public Overridable String[] GetFolderInfo(String Path);
```

Visual Basic

```
Public Overridable Function GetFolderInfo(ByVal Path As String) As String()
```

Parameters

Parameters	Description
ByVal Path As String	The folder's path.

Returns

String()

Folder.GetPath Method

Gets the path of a folder in the Workarea by using the folder's ID.

C#

```
public String GetPath(Long Id);
```

Visual Basic

```
Public Function GetPath(ByVal Id As Long) As String
```

Parameters

Parameters	Description
ByVal Id As Long	The folder's ID.

Returns

Returns the folder path as a string. For example: `\~SiteRoot~\Products\Medication Management`.

Folder.IsFolderHidden Method

Returns flag indicating if the folder is hidden.

C#

```
public Boolean IsFolderHidden(Long folderId);
```

Visual Basic

```
Public Function IsFolderHidden(ByVal folderId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal folderId As Long	The ID of the folder that you are checking.

Returns

Returns True if the folder is hidden, otherwise returns False.

Folder.RenameFolder Method

Rename the folder with new name.

C#

```
public Boolean RenameFolder(Long cId, String title);
```

Visual Basic

```
Public Function RenameFolder(ByVal cId As Long, ByVal title As String) As Boolean
```

Parameters

Parameters	Description
ByVal cId As Long	The folder's ID.
ByVal title As String	The new folder's name.

Returns

Returns False if successfully renamed. Otherwise, it returns True.

UpdateFolder Method

This is the overview for the UpdateFolder method overload.

Overload List

	Name	Description
	Folder.UpdateFolder (FolderData) (see page 694)	Updates the CMS folder's information.
	Folder.UpdateFolder (FolderRequest) (see page 695)	Updates the CMS folder's information.

Folder.UpdateFolder Method (FolderData)

Updates the CMS folder's information.

C#

```
public UpdateFolder(FolderData request);
```

Visual Basic

```
Public Sub UpdateFolder(ByVal request As FolderData)
```

Parameters

Parameters	Description
ByVal request As FolderData	The Folder (see page 683)'s properties as a FolderData object.

Folder.UpdateFolder Method (FolderRequest)

Updates the CMS folder's information.

C#

```
public UpdateFolder(FolderRequest request);
```

Visual Basic

```
Public Sub UpdateFolder(ByVal request As FolderRequest)
```

Parameters

Parameters	Description
ByVal request As FolderRequest	The Folder (see page 683)'s properties as a FolderRequest (see page 1615) object.

Folder.UpdateFolderApprovalMethod Method

Updates the ApprovalMethod of a folder. The ApprovalMethod gets or set whether or not to force all approvers to approve the content that is added to the folder. Before using this method, you must break permission inheritance for the folder by setting the FolderRequest (see page 1615) object's BreakInheritButton property to True.

C#

```
public UpdateFolderApprovalMethod(FolderData request);
```

Visual Basic

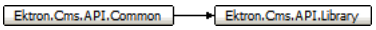
```
Public Sub UpdateFolderApprovalMethod(ByVal request As FolderData)
```

Parameters

Parameters	Description
ByVal request As FolderData	The Folder (see page 683)'s properties as a FolderData object.

Library Class

Class Hierarchy



C#

```
public class Library : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Library
Inherits Ektron.Cms.API.Common
```

File

Library.vb

Description

This is class Ektron.Cms.API.Library.

Library Methods

The methods of the Library class are listed here.

Public Methods

	Name	Description
	AddLibraryItem (see page 696)	Adds a library item's information to the database. It does not actually upload the item to the library.
	AddLoadBalanceItem (see page 696)	Adds a load balance path for files and images.
	DeleteLibraryItem (see page 697)	Deletes the library item by extension.
	DeleteLibraryItemById (see page 697)	Deletes the library items from the system.
	DeleteLoadBalanceItem (see page 697)	Deletes the load balance item from the system.
	GetAllChildLibItems (see page 697)	Loads all of the library items by type. Using the OrderBy parameter allows you to sort the library items. The currentPageNum, pageSize and totalPages parameters are used with paging. Setting currentPageNum and pageSize to zero causes the API (see page 669) To retrieve all items on a single page. **Note**: The caller of API (see page 669) must actually call the API (see page 669) with the correct page number to implement paging. The API (see page 669) does not provide any inherent UI for paging purposes.
	GetAllLoadBalancePaths (see page 699)	This is the overview for the GetAllLoadBalancePaths method overload.
	GetLibraryItem (see page 700)	This is the overview for the GetLibraryItem method overload.
	GetLibrarySettings (see page 701)	Loads the library settings.
	GetLibraryTypes (see page 702)	Loads the library types.
	GetLoadBalancePath (see page 702)	Gets the load balance path by ID.
	InternalLibrarySearch (see page 702)	Searches library items.
	UpdateLibrary (see page 703)	Updates a library item.
	UpdateLibrarySettings (see page 703)	Update the library settings.
	UpdateLoadBalanceSettings (see page 703)	Updates the load balance settings.
	UpdateQlinkTemplates (see page 703)	Updates the Quicklink template.

Library.AddLibraryItem Method

Adds a library item's information to the database. It does not actually upload the item to the library.

C#

```
public Long AddLibraryItem(ref LibraryData libraryData);
```

Visual Basic

```
Public Function AddLibraryItem(ByRef libraryData As LibraryData) As Long
```

Parameters

Parameters	Description
ByRef libraryData As LibraryData	Details of the new library item

Library.AddLoadBalanceItem Method

Adds a load balance path for files and images.

C#

```
public AddLoadBalanceItem(String Type, Collection pagedata);
```

Visual Basic

```
Public Sub AddLoadBalanceItem(ByVal Type As String, ByVal pagedata As Collection)
```

Parameters

Parameters	Description
ByVal Type As String	The path type, either images or files.
ByVal pagedata As Collection	Load balance item values for path and relative.

Library.DeleteLibraryItem Method

Deletes the library item by extension.

C#

```
public DeleteLibraryItem(Collection pagedata);
```

Visual Basic

```
Public Sub DeleteLibraryItem(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
ByVal pagedata As Collection	Library (see page 695) item data

Library.DeleteLibraryItemById Method

Deletes the library items from the system.

C#

```
public DeleteLibraryItemById(Collection pagedata);
```

Visual Basic

```
Public Sub DeleteLibraryItemById(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
ByVal pagedata As Collection	Library (see page 695) item data

Library.DeleteLoadBalanceItem Method

Deletes the load balance item from the system.

C#

```
public DeleteLoadBalanceItem(Long Id);
```

Visual Basic

```
Public Sub DeleteLoadBalanceItem(ByVal Id As Long)
```

Parameters

Parameters	Description
ByVal Id As Long	The load balance item ID

Library.GetAllChildLibItems Method

Loads all of the library items by type. Using the OrderBy parameter allows you to sort the library items. The currentPageNum, pageSize and totalPages parameters are used with paging. Setting currentPageNum and pageSize to zero causes the API (see page 669) To retrieve all items on a single page.

Note: The caller of API (see page 669) must actually call the API (see page 669) with the correct page number to implement paging. The API (see page 669) does not provide any inherent UI for paging purposes.

C#

```
public LibraryData[] GetAllChildLibItems(String Type, Long ParentId, String OrderBy, int
currentPageNum, int pageSize, ref int totalPages);
```

Visual Basic

```
Public Function GetAllChildLibItems(ByVal Type As String, ByVal ParentId As Long, ByVal
OrderBy As String, ByVal currentPageNum As Integer, ByVal pageSize As Integer, ByRef
totalPages As Integer) As LibraryData()
```

Parameters

Parameters	Description
ByVal Type As String	The library item's type. (Required) Use one of the following: • images • files • hyperlinks • quicklinks • forms
ByVal ParentId As Long	The folder's ID.
ByVal OrderBy As String	The order to load the library items. You can pass: • librarytitle • libraryid • libraryfilename • date
ByVal currentPageNum As Integer	The current page number to display.
ByVal pageSize As Integer	The page size set by the caller of the API (see page 669).
ByRef totalPages As Integer	The total number of pages required to display all of the items.

Returns

Ektron.Cms.LibraryData()

Example

The following example shows how to create a Web page that displays a graphical list of all images stored in the library for a specific folder in the Workarea. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetAllChildLibItems method. This method uses the InternalAdmin to retrieve the images. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a Label with the following information.
```
<asp:Label ID="Label6" runat="server" Text="Folder ID:" Width="76px"></asp:Label>
```
2. Add a text box for the Folder ID.
```
<asp:TextBox ID="txtFolderId" runat="server" Width="55px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder ID box.
```
<asp:RequiredFieldValidator ID="rfvId" runat="server" ControlToValidate="txtFolderId"
    ErrorMessage="Please enter a folder ID"></asp:RequiredFieldValidator>
```
4. Add a couple of breaks for spacing purposes.
```
<br /><br />
```
5. Add a submit button. We will set the button click event in the code behind.
```
<asp:Button ID="btnGet" runat="server" Text="Get Images" Width="74px" />
```
6. Add two more breaks for spacing purposes.
```
<br /><br />
```
7. Add a label to display the list of images.
```
<asp:Label ID="lblResult" runat="server"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
        Protected Sub btnGet_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGet.Click
```

2. Check to make sure the folder ID is numeric.

```
            If Not IsNumeric(txtFolderId.Text) Then
                lblResult.Text = "Invalid Folder ID"
                Exit Sub
            End If
```

3. Create an object for the library API (see page 669).

```
            Dim libraryApi As New Ektron.Cms.API.Library
```

4. Create an array to contain the library data.

```
            Dim data As Ektron.Cms.LibraryData()
```

5. Create a Try/Catch around the GetAllChildLibItems method. This allows you to catch any exceptions and helps when debugging.

```
            Try
                data = libraryApi.GetAllChildLibItems("images", CInt(txtFolderId.Text), "",
0, 0, 0)
```

6. Check to make sure the library has images for the folder.

```
                If (Not data Is Nothing) AndAlso (data.Length > 0) Then
```

7. If so, create an object for each library item.

```
                    Dim item As Ektron.Cms.LibraryData
```

8. Create an empty string.

```
                    Dim str As String = ""
```

9. Iterate through each item and display the image with the image title as the alt text.

```
                     For Each item In data
                         str &= "<img src=""" & item.FileName & """ alt=""" & item.Title &
"""/>" & "<br/>" & "<br/>"
                         lblResult.Text = str
                     Next
```

10. If the library contains no images for the folder, display a message.

```
                 Else
                     lblResult.Text = "No images found"
                 End If
```

11. Display a message if there is an error.

```
            Catch ex As Exception
                lblResult.Text = ex.Message
            End Try

        End Sub
```

GetAllLoadBalancePaths Method

This is the overview for the GetAllLoadBalancePaths method overload.

Description

Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.

Overload List

	Name	Description
	Library.GetAllLoadBalancePaths (Long, String) (see page 700)	Returns an array of LoadBalanceData.
	Library.GetAllLoadBalancePaths (String) (see page 700)	Returns all the load balance paths as a array of LoadBalanceData.

Library.GetAllLoadBalancePaths Method (Long, String)

Returns an array of LoadBalanceData.

C#

```
public LoadBalanceData[] GetAllLoadBalancePaths(Long Id, String Type);
```

Visual Basic

```
Public Function GetAllLoadBalancePaths(ByVal Id As Long, ByVal Type As String) As
LoadBalanceData()
```

Parameters

Parameters	Description
ByVal Id As Long	Load balance ID
ByVal Type As String	Use images or files

Returns

Ektron.Cms.LoadBalanceData()

Library.GetAllLoadBalancePaths Method (String)

Returns all the load balance paths as a array of LoadBalanceData.

C#

```
public LoadBalanceData[] GetAllLoadBalancePaths(String Type);
```

Visual Basic

```
Public Function GetAllLoadBalancePaths(ByVal Type As String) As LoadBalanceData()
```

Parameters

Parameters	Description
ByVal Type As String	Type (images or files)

Returns

Ektron.Cms.LoadBalanceData()

GetLibraryItem Method

This is the overview for the GetLibraryItem method overload.

Description

Loads the library item's data. This method is overloaded.

Overload List

	Name	Description
	Library.GetLibraryItem (Long) (see page 701)	Loads the library item data.
	Library.GetLibraryItem (Long, Long) (see page 701)	Loads the library item's data.

Library.GetLibraryItem Method (Long)

Loads the library item data.

C#

```
public LibraryData GetLibraryItem(Long LibID);
```

Visual Basic

```
Public Function GetLibraryItem(ByVal LibID As Long) As LibraryData
```

Parameters

Parameters	Description
ByVal LibID As Long	The library ID

Returns

Ektron.Cms.LibraryData

Library.GetLibraryItem Method (Long, Long)

Loads the library item's data.

C#

```
public LibraryData GetLibraryItem(Long Id, Long ParentId);
```

Visual Basic

```
Public Function GetLibraryItem(ByVal Id As Long, ByVal ParentId As Long) As LibraryData
```

Parameters

Parameters	Description
ByVal Id As Long	The library's ID
ByVal ParentId As Long	The folder's ID

Returns

Ektron.Cms.LibraryData

Library.GetLibrarySettings Method

Loads the library settings.

C#

```
public LibraryConfigData GetLibrarySettings(Optional ByVal Long Id);
```

Visual Basic

```
Public Function GetLibrarySettings(Optional ByVal Id As Long = -1) As LibraryConfigData
```

Parameters

Parameters	Description
Optional ByVal Id As Long = -1	The folder's ID

Returns

Ektron.Cms.LibraryConfigData

Library.GetLibraryTypes Method

Loads the library types.

C#

```
public LibraryTypeData[] GetLibraryTypes(Long Id, String OrderBy);
```

Visual Basic

```
Public Function GetLibraryTypes(ByVal Id As Long, ByVal OrderBy As String) As LibraryTypeData()
```

Parameters

Parameters	Description
ByVal Id As Long	The folder's ID
ByVal OrderBy As String	Order by clause

Returns

Object

Library.GetLoadBalancePath Method

Gets the load balance path by ID.

C#

```
public LoadBalanceData GetLoadBalancePath(Long Id);
```

Visual Basic

```
Public Function GetLoadBalancePath(ByVal Id As Long) As LoadBalanceData
```

Parameters

Parameters	Description
ByVal Id As Long	the load balance path's ID

Returns

Ektron.Cms.LoadBalanceData

Library.InternalLibrarySearch Method

Searches library items.

C#

```
public SearchResultData[] InternalLibrarySearch(Collection pagedata, Optional ByVal
Ektron.Cms.Common.SearchAssetRequest sar);
```

Visual Basic

```
Public Function InternalLibrarySearch(ByVal pagedata As Collection, Optional ByVal sar As
Ektron.Cms.Common.SearchAssetRequest = Nothing) As SearchResultData()
```

Parameters

Parameters	Description
ByVal pagedata As Collection	Search (see page 899) criteria
Optional ByVal sar As Ektron.Cms.Common.SearchAssetRequest = Nothing	Search (see page 899) asset criteria

Returns

Ektron.Cms.SearchResultData()

Library.UpdateLibrary Method

Updates a library item.

C#

```
public UpdateLibrary(Collection pagedata);
```

Visual Basic

```
Public Sub UpdateLibrary(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
ByVal pagedata As Collection	Library (see page 695) item data

Library.UpdateLibrarySettings Method

Update the library settings.

C#

```
public UpdateLibrarySettings(Collection pagedata);
```

Visual Basic

```
Public Sub UpdateLibrarySettings(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
ByVal pagedata As Collection	Library (see page 695) settings data

Library.UpdateLoadBalanceSettings Method

Updates the load balance settings.

C#

```
public UpdateLoadBalanceSettings(String Id, Collection pagedata);
```

Visual Basic

```
Public Sub UpdateLoadBalanceSettings(ByVal Id As String, ByVal pagedata As Collection)
```

Parameters

Parameters	Description
ByVal Id As String	The load balance data
ByVal pagedata As Collection	The load balance items

Library.UpdateQlinkTemplates Method

Updates the Quicklink template.

C#

```
public UpdateQlinkTemplates(Collection pagedata);
```

Visual Basic

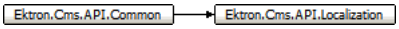
```
Public Sub UpdateQlinkTemplates(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
ByVal pagedata As Collection	Quicklink template data

Localization Class

Class Hierarchy



C#

```
public class Localization : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Localization
Inherits Ektron.Cms.API.Common
```

File

Localization.vb

Description

This is class Ektron.Cms.API.Localization.

Localization Methods

The methods of the Localization class are listed here.

Public Methods

	Name	Description
	GetAddViewLanguage (see page 705)	Returns an array of all the active content languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") for the specified content ID or whether it does not exist in that language (Type="ADD").
	GetAddViewLanguageForAllContent (see page 705)	Returns an array of all the active content languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one content block or whether no content exists in that language (Type="ADD").
	GetAddViewLanguageForMenus (see page 705)	Returns an array of all the active content languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one menu or whether no menus exist in that language (Type="ADD").
	GetContentItems (see page 706)	Returns a list of content items (i.e., XLIFF skeletons) associated with a given job.
	GetFile (see page 706)	Returns the specified file record.
	GetFiles (see page 706)	Returns a list of files associated with a given job.
	GetFilesInZipFile (see page 707)	Returns a list of files in a given ZIP file.
	GetFlagUrl (see page 707)	This is the overview for the GetFlagUrl method overload.
	GetInterchangeFile (see page 709)	Returns the interchange (i.e., XLIFF) file record specified by ID.
	GetInterchangeFiles (see page 709)	Returns a list of interchange files (i.e., XLIFF files) associated with a given content item (i.e., XLIFF Skeleton).
	GetInterchangeFilesInZipFile (see page 709)	Returns a list of interchange files (i.e., XLIFF files) contained within a given ZIP file.
	GetJobs (see page 710)	This is the overview for the GetJobs method overload.
	GetLocalizationUrl (see page 711)	Returns a URL to the base directory where localization files are stored. `For example, /~SiteRoot~/uploadedfiles/localization.`

	GetTranslationUploadDirectory (see page 711)	Returns the server directory path to upload XLIFF and ZIP files for importing. For example, `C:\inetpub\wwwroot\~SiteRoot~\workarea\uploadedfiles\localization`. See Also: StartImportTranslation.
	StartExportContentForTranslation (see page 711)	Starts the process of exporting the content specified by ListOfContentIDs. The status of the process may be monitored by visiting localizationjobs.aspx, which uses methods in this class. The translatable text and documents are exported as XLIFF files compressed in ZIP files. The XLIFF files are suitable for sending to a translator. Once translated, the XLIFF files should be imported using the StartImportTranslation method.
	StartExportFolderForTranslation (see page 712)	Starts the process of exporting all the content specified by `ContentType` in the specified folder. Optionally, all the content in the subfolders may be processed as well. The status of the process may be monitored by visiting localizationjobs.aspx, which uses methods in this class. The translatable text and documents are exported as XLIFF files compressed in ZIP files. The XLIFF files are suitable for sending to a translator. Once translated, the XLIFF files should be imported using the StartImportTranslation.
	StartExportMenusForTranslation (see page 712)	Starts the process of exporting all menus. The status of the process may be monitored by visiting localizationjobs.aspx, which uses methods in this class. The translatable text is exported as an XLIFF file and compressed in a ZIP file. The XLIFF file is suitable for sending to a translator. Once translated, the XLIFF file should be imported using the StartImportTranslation method.
	StartImportTranslation (see page 713)	Starts the process of importing XLIFF files that have been translated. The files should be uploaded to server directory specified by GetTranslationUploadDirectory. Files may be XLIFF files (.xlf extension) or XLIFF files that are compressed in ZIP (.zip extension) files.

Localization.GetAddViewLanguage Method

Returns an array of all the active content languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") for the specified content ID or whether it does not exist in that language (Type="ADD").

C#

```
public LanguageData[] GetAddViewLanguage(Long Id);
```

Visual Basic

```
Public Function GetAddViewLanguage(ByVal Id As Long) As LanguageData()
```

Parameters

Parameters	Description
ByVal Id As Long	Content (see page 822) ID

Returns

Ektron.Cms.LanguageData()

Localization.GetAddViewLanguageForAllContent Method

Returns an array of all the active content languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one content block or whether no content exists in that language (Type="ADD").

C#

```
public LanguageData[] GetAddViewLanguageForAllContent();
```

Visual Basic

```
Public Function GetAddViewLanguageForAllContent() As LanguageData()
```

Returns

LanguageData()

Remarks

Only Id, Name, BrowserCode and Type are populated.

Localization.GetAddViewLanguageForMenus Method

Returns an array of all the active content languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one menu or whether no menus exist in that language (Type="ADD").

C#

```
public LanguageData[] GetAddViewLanguageForMenus();
```

Visual Basic

```
Public Function GetAddViewLanguageForMenus() As LanguageData()
```

Returns

LanguageData()

Remarks

Only Id, Name, BrowserCode and Type are populated.

Localization.GetContentItems Method

Returns a list of content items (i.e., XLIFF skeletons) associated with a given job.

C#

```
public DS.LocalizationSkeletonDataTable GetContentItems(Long JobID);
```

Visual Basic

```
Public Function GetContentItems(ByVal JobID As Long) As DS.LocalizationSkeletonDataTable
```

Parameters

Parameters	Description
ByVal JobID As Long	Specifies the job.

Returns

A LocalizationJobDataSet.LocalizationSkeletonDataTable of content (i.e., skeleton) items.

Localization.GetFile Method

Returns the specified file record.

C#

```
public DS.LocalizationJobFileRow GetFile(Long FileID);
```

Visual Basic

```
Public Function GetFile(ByVal FileID As Long) As DS.LocalizationJobFileRow
```

Parameters

Parameters	Description
ByVal FileID As Long	Specifies a file (e.g., ZIP file)

Returns

A LocalizationJobDataSet.LocalizationJobFileRow" record of a job file or Nothing if not found.

Localization.GetFiles Method

Returns a list of files associated with a given job.

C#

```
public DS.LocalizationJobFileDataTable GetFiles(Long JobID);
```

Visual Basic

```
Public Function GetFiles(ByVal JobID As Long) As DS.LocalizationJobFileDataTable
```

Parameters

Parameters	Description
ByVal JobID As Long	Specifies the job.

Returns

A LocalizationJobDataSet.LocalizationJobFileDataTable of files.

Remarks

Files are typically ZIP files for export jobs and ZIP and XLIFF files for import jobs.

Localization.GetFilesInZipFile Method

Returns a list of files in a given ZIP file.

C#

```
public DS.LocalizationJobFileDataTable GetFilesInZipFile(Long FileID);
```

Visual Basic

```
Public Function GetFilesInZipFile(ByVal FileID As Long) As DS.LocalizationJobFileDataTable
```

Parameters

Parameters	Description
ByVal FileID As Long	Specifies a file (e.g., ZIP file)

Returns

A LocalizationJobDataSet.LocalizationJobFileDataTable of files.

Remarks

Files are typically ZIP and XLIFF files for import jobs. The export process usually will not generate ZIP files with other ZIP files in them.

GetFlagUrl Method

This is the overview for the GetFlagUrl method overload.

Description

Returns a URL to a small image of a national flag for the country and language specified by LanguageID or Data. This method is overloaded.

Overload List

	Name	Description
	Localization.GetFlagUrl (Integer) (see page 708)	Returns a URL to a small image of a national flag for the country and language specified by LanguageID.
	Localization.GetFlagUrl (LanguageData) (see page 708)	Returns a URL to a small image of a national flag for the country and language specified by Data.
	Localization.GetFlagUrl (LanguageDataSet.LanguageRow) (see page 708)	Returns a URL to a small image of a national flag for the country and language specified by Data.

Localization.GetFlagUrl Method (Integer)

Returns a URL to a small image of a national flag for the country and language specified by LanguageID.

C#

```
public String GetFlagUrl(int LanguageID);
```

Visual Basic

```
Public Function GetFlagUrl(ByVal LanguageID As Integer) As String
```

Parameters

Parameters	Description
ByVal LanguageID As Integer	Specifies the locale, that is, the language and country

Returns

A URL to an image of a national flag.

Remarks

Often, only the language is of concern and the flag for the default country will be returned.

Localization.GetFlagUrl Method (LanguageData)

Returns a URL to a small image of a national flag for the country and language specified by Data.

C#

```
public String GetFlagUrl(LanguageData Data);
```

Visual Basic

```
Public Function GetFlagUrl(ByVal Data As LanguageData) As String
```

Parameters

Parameters	Description
ByVal Data As LanguageData	Specifies the language data.

Returns

A URL to an image of a national flag.

Remarks

Often, only the language is of concern and the flag for the default country will be returned.

Localization.GetFlagUrl Method (LanguageDataSet.LanguageRow)

Returns a URL to a small image of a national flag for the country and language specified by Data.

C#

```
public String GetFlagUrl(LanguageDataSet.LanguageRow Data);
```

Visual Basic

```
Public Function GetFlagUrl(ByVal Data As LanguageDataSet.LanguageRow) As String
```

Parameters

Parameters	Description
ByVal Data As LanguageDataSet.LanguageRow	Specifies the language data.

Returns

A URL to an image of a national flag.

Remarks

Often, only the language is of concern and the flag for the default country will be returned.

Localization.GetInterchangeFile Method

Returns the interchange (i.e., XLIFF) file record specified by ID.

C#

```
public DS.LocalizationTargetRow GetInterchangeFile(Long ID);
```

Visual Basic

```
Public Function GetInterchangeFile(ByVal ID As Long) As DS.LocalizationTargetRow
```

Parameters

Parameters	Description
ByVal ID As Long	Specifies the interchange (i.e., XLIFF) file record.

Returns

A LocalizationJobDataSet.LocalizationTargetRow data record of the selected XLIFF file or Nothing if it does not exist.

Localization.GetInterchangeFiles Method

Returns a list of interchange files (i.e., XLIFF files) associated with a given content item (i.e., XLIFF Skeleton).

C#

```
public DS.LocalizationTargetDataTable GetInterchangeFiles(Long ItemID);
```

Visual Basic

```
Public Function GetInterchangeFiles(ByVal ItemID As Long) As DS.LocalizationTargetDataTable
```

Parameters

Parameters	Description
ByVal ItemID As Long	Specifies the content item (i.e., XLIFF skeleton).

Returns

A LocalizationJobDataSet.LocalizationTargetDataTable of XLIFF files.

Remarks

When exporting, XLIFF files for each target language are generated from a skeleton. When importing, an XLIFF file is associated with the skeleton content item it was generated from (the original language) and a new skeleton content item generated by merging the XLIFF with the original skeleton to produce a skeleton in the target language. A skeleton consists of the content, typically the main HTML content block, a summary (HTML), title, meta data, etc.

Localization.GetInterchangeFilesInZipFile Method

Returns a list of interchange files (i.e., XLIFF files) contained within a given ZIP file.

C#

```
public DS.LocalizationTargetDataTable GetInterchangeFilesInZipFile(Long FileID);
```

Visual Basic

```
Public Function GetInterchangeFilesInZipFile(ByVal FileID As Long) As
DS.LocalizationTargetDataTable
```

Parameters

Parameters	Description
ByVal FileID As Long	Specifies the ZIP file.

Returns

A LocalizationJobDataSet.LocalizationTargetDataTable of XLIFF files.

Remarks

XLIFF is a standard to interchange localization information for translation of text. The XLIFF files are compressed together into one or more ZIP files for convenience.

GetJobs Method

This is the overview for the GetJobs method overload.

Description

Gets a list of top level jobs or a list of the sub jobs of the given job. This method is overloaded.

Overload List

	Name	Description
	Localization.GetJobs () (see page 710)	Returns a list of the top-level jobs, export and import jobs. The jobs listed depend on the user id, unauthenticated users (e.g., translators) see only publicly available jobs (e.g., ZIP files for download). Administrators can view all jobs, non-administrators may only view jobs they created. This function is overloaded. See Also: GetJobs (Long)
	Localization.GetJobs (Long) (see page 710)	Returns a list of the sub jobs of the given job. This function is overloaded. See Also: GetJobs

Localization.GetJobs Method ()

Returns a list of the top-level jobs, export and import jobs. The jobs listed depend on the user id, unauthenticated users (e.g., translators) see only publicly available jobs (e.g., ZIP files for download). Administrators can view all jobs, non-administrators may only view jobs they created.

This function is overloaded. See Also: GetJobs (Long)

C#

```
public DS.LocalizationJobDataTable GetJobs();
```

Visual Basic

```
Public Function GetJobs() As DS.LocalizationJobDataTable
```

Returns

A LocalizationJobDataSet.LocalizationJobDataTable of top-level jobs.

Localization.GetJobs Method (Long)

Returns a list of the sub jobs of the given job.

This function is overloaded. See Also: GetJobs (see page 710)

C#

```
public DS.LocalizationJobDataTable GetJobs(Long ParentJobID);
```

Visual Basic

```
Public Function GetJobs(ByVal ParentJobID As Long) As DS.LocalizationJobDataTable
```

Parameters

Parameters	Description
ByVal ParentJobID As Long	Specifies the job.

Returns

A LocalizationJobDataSet.LocalizationJobDataTable data table of jobs.

Remarks

Localization.GetLocalizationUrl Method

Returns a URL to the base directory where localization files are stored. For example, `/~SiteRoot~/uploadedfiles/localization`.

C#

```
public String GetLocalizationUrl();
```

Visual Basic

```
Public Function GetLocalizationUrl() As String
```

Returns

A URL to the base localization directory.

Localization.GetTranslationUploadDirectory Method

Returns the server directory path to upload XLIFF and ZIP files for importing. For example, `C:\inetpub\wwwroot\~SiteRoot~\workarea\uploadedfiles\localization`.

See Also: StartImportTranslation (see page 713).

C#

```
public String GetTranslationUploadDirectory();
```

Visual Basic

```
Public Function GetTranslationUploadDirectory() As String
```

Returns

The server directory path (as a String) to store XLIFF and ZIP files that are to be imported.

Remarks

This directory path should not be displayed on a web page.

Localization.StartExportContentForTranslation Method

Starts the process of exporting the content specified by ListOfContentIDs. The status of the process may be monitored by visiting localizationjobs.aspx, which uses methods in this class. The translatable text and documents are exported as XLIFF files compressed in ZIP files. The XLIFF files are suitable for sending to a translator. Once translated, the XLIFF files should be imported using the StartImportTranslation (see page 713) method.

C#

```
public StartExportContentForTranslation(String ListOfContentIDs, Optional ByVal String
```

```
ListOfLanguageIDs);
```

Visual Basic

```
Public Sub StartExportContentForTranslation(ByVal ListOfContentIDs As String, Optional ByVal
ListOfLanguageIDs As String = "")
```

Parameters

Parameters	Description
ByVal ListOfContentIDs As String	Comma-delimited list of content IDs. e.g., "11,12"
Optional ByVal ListOfLanguageIDs As String = ""	Comma-delimited list of language IDs. e.g., "1031,1036"

Remarks

To export all content in a folder, use StartExportFolderForTranslation (see page 712) (Long Integer Boolean String). See http://www.xliff.org for more information about XLIFF.

Localization.StartExportFolderForTranslation Method

Starts the process of exporting all the content specified by `ContentType` in the specified folder. Optionally, all the content in the subfolders may be processed as well. The status of the process may be monitored by visiting localizationjobs.aspx, which uses methods in this class. The translatable text and documents are exported as XLIFF files compressed in ZIP files. The XLIFF files are suitable for sending to a translator. Once translated, the XLIFF files should be imported using the StartImportTranslation (see page 713).

C#

```
public StartExportFolderForTranslation(Long FolderID, int ContentType, Boolean
IncludeSubfolders, Optional ByVal String ListOfLanguageIDs);
```

Visual Basic

```
Public Sub StartExportFolderForTranslation(ByVal FolderID As Long, ByVal ContentType As
Integer, ByVal IncludeSubfolders As Boolean, Optional ByVal ListOfLanguageIDs As String = "")
```

Parameters

Parameters	Description
ByVal FolderID As Long	Specifies the folder containing the content to export.
ByVal ContentType As Integer	Specifies which type or types of content to export.
ByVal IncludeSubfolders As Boolean	Specifies whether to export just the specified folder or all subfolders as well.
Optional ByVal ListOfLanguageIDs As String = ""	Comma-delimited list of language IDs. e.g., "1031,1036"

Remarks

To export just selected content, use StartExportFolderForTranslation (Long Integer Boolean String)

See http://www.xliff.org for more information about XLIFF.

Localization.StartExportMenusForTranslation Method

Starts the process of exporting all menus. The status of the process may be monitored by visiting localizationjobs.aspx, which uses methods in this class. The translatable text is exported as an XLIFF file and compressed in a ZIP file. The XLIFF file is suitable for sending to a translator. Once translated, the XLIFF file should be imported using the StartImportTranslation (see page 713) method.

C#

```
public StartExportMenusForTranslation(Optional ByVal String ListOfLanguageIDs);
```

Visual Basic

```
Public Sub StartExportMenusForTranslation(Optional ByVal ListOfLanguageIDs As String = "")
```

Parameters

Parameters	Description
Optional ByVal ListOfLanguageIDs As String = ""	Comma-delimited list of language IDs. e.g., "1031,1036"

Remarks

See http://www.xliff.org for more information about XLIFF.

Localization.StartImportTranslation Method

Starts the process of importing XLIFF files that have been translated. The files should be uploaded to server directory specified by GetTranslationUploadDirectory (see page 711). Files may be XLIFF files (.xlf extension) or XLIFF files that are compressed in ZIP (.zip extension) files.

C#

```
public StartImportTranslation();
```

Visual Basic

```
Public Sub StartImportTranslation()
```

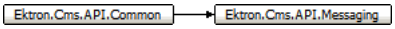
Remarks

See http://www.xliff.org for more information about XLIFF.

Messaging Class

The Messaging class allows you to work with Private Message Objects.

Class Hierarchy



C#

```
public class Messaging : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Messaging
Inherits Ektron.Cms.API.Common
```

File

Messaging.vb

Messaging Methods

The methods of the Messaging class are listed here.

Public Methods

	Name	Description
	GetUnreadMessageCountForUser (see page 713)	Gets the total unread messages for a given user ID.

Messaging.GetUnreadMessageCountForUser Method

Gets the total unread messages for a given user ID.

C#

```
public int GetUnreadMessageCountForUser(Long userId);
```

Visual Basic

```
Public Function GetUnreadMessageCountForUser(ByVal userId As Long) As Integer
```

Parameters

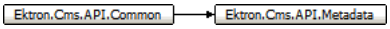
Parameters	Description
ByVal userId As Long	ID of the user.

Returns

Integer count of unread messages for a user.

Metadata Class

Class Hierarchy



C#

```
public class Metadata : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Metadata
Inherits Ektron.Cms.API.Common
```

File

Metadata.vb

Description

This is class Ektron.Cms.API.Metadata.

Metadata Methods

The methods of the Metadata class are listed here.

Public Methods

	Name	Description
	AddMetaDataType (see page 714)	This is the overview for the AddMetaDataType method overload.
	DeleteMetadataType (see page 717)	Deletes a metadata type from the system.
	GetContentMetadataList (see page 717)	This is the overview for the GetContentMetadataList method overload.
	GetIntegratedSearchFolder (see page 719)	Loads all custom folder data.
	GetIntegratedSearchFolders (see page 719)	Loads all the custom folders.
	GetMetadataByFolder (see page 720)	Returns a collection of ContentMetadata assigned to a folder.
	GetMetadataType (see page 720)	Loads all content metadata.
	GetMetaDataTypes (see page 720)	Loads all the metadata types.
	UpdateMetadataType (see page 721)	Updates the metadata type.

AddMetaDataType Method

This is the overview for the AddMetaDataType method overload.

Overload List

	Name	Description
	Metadata.AddMetaDataType (Collection) (see page 715)	Adds a new metadata type to the system.
	Metadata.AddMetaDataType (ContentMetaData) (see page 715)	Adds a new metadata type to the system.

Metadata.AddMetaDataType Method (Collection)

Adds a new metadata type to the system.

C#

```
public AddMetaDataType(Collection pagedata);
```

Visual Basic

```
Public Sub AddMetaDataType(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
ByVal pagedata As Collection	Metadata (see page 714) items as value key pair.

Metadata.AddMetaDataType Method (ContentMetaData)

Adds a new metadata type to the system.

C#

```
public AddMetaDataType(ref ContentMetaData metaDataItem);
```

Visual Basic

```
Public Sub AddMetaDataType(ByRef metaDataItem As ContentMetaData)
```

Parameters

Parameters	Description
ByRef metaDataItem As ContentMetaData	ContentMetadata object containing metadata definition details.

Example

The following example shows how to create a Web page from which you can add a new metadata definition. This example uses some standard drag and drop controls and a small section of VB code utilizing the AddMetaDataType(ByRef metaDataItem As ContentMetaData) method. This method uses the InternalAdmin to add the metadata definition. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Create a table with four rows and three columns.
2. In the first row, first column, add a label that shows where to enter the name of the metadata definition.
```
<asp:Label ID="Label1" runat="server" Text="Name: "></asp:Label>
```
3. In the first row, second column, add a text box for the user to enter the metadata definition name.
```
<asp:TextBox ID="txtMetaName" runat="server" Width="86px"></asp:TextBox>
```
4. In the first row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the name box.
```
<asp:RequiredFieldValidator ID="rfvMetaName" runat="server" ControlToValidate="txtMetaName"
  ErrorMessage="Please enter metadata definition name"
Width="237px"></asp:RequiredFieldValidator>
```
5. In the second row, first column, add a label that show where to select the type.
```
<asp:Label ID="Label4" runat="server" Text="Type:" Width="80px"></asp:Label>
```
6. In the second row, second column, add a drop down list for the metadata tag type. We will populate the drop down list in the

code behind.

```
<asp:DropDownList ID="ddlMetaTagType" runat="server" Width="194px"></asp:DropDownList>
```

7. Leave the second row, third column empty.
8. In the thrid row, first column, add a label for the Editable option.

```
<asp:Label ID="Label7" runat="server" Text="Editable: " Width="76px"></asp:Label>
```

9. In the third row, second column, add a check box to decide whether a user can edit the contents of the metadata when creating or editing the metadata's content.

```
<asp:CheckBox ID="chkEditable" runat="server" Checked="True" />
```

10. Leave the third row, third column empty.
11. In the fourth row, first column, add a label for the Display in Ektron Explorer option.

```
<asp:Label ID="Label2" runat="server" Text="Display in Ektron Explorer"
Width="119px"></asp:Label>
```

12. In the fourth row, second column, add a check box to decide whether the metadata definition should be displayed in Ektron Explorer.

```
<asp:CheckBox ID="chkDisplayEE" runat="server" />
```

13. Leave the fourth row, third column empty.
14. After the table, add a break and then add a button to add the Metadata Definition.

```
<br />
<asp:Button ID="btnAdd" runat="server" Text="Add Metatype" Width="98px" />
```

15. Add two breaks for spacing purposes.

```
<br /><br />
```

16. Add a results label to display whether the Metadata Definition was added and if so, its ID.

```
<asp:Label ID="lblResult" runat="server" Width="549px" Height="25px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a page load event.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
```

2. In the page load event, populate the drop down list with the available metadata definition types. These are the only types available. Make sure you add the numeric value. This corresponds to metadata definition type in the database.

```
    If (Not IsPostBack()) Then
        ddlMetaTagType.Items.Add(New ListItem("Searchable Property", "100"))
        ddlMetaTagType.Items.Add(New ListItem("Meta Tag", "1"))
        ddlMetaTagType.Items.Add(New ListItem("HTML Tag", "0"))
        ddlMetaTagType.Items.Add(New ListItem("Collection Selector", "2"))
        ddlMetaTagType.Items.Add(New ListItem("Content Selector", "4"))
        ddlMetaTagType.Items.Add(New ListItem("File Selector", "7"))
        ddlMetaTagType.Items.Add(New ListItem("Hyperlink Selector", "6"))
        ddlMetaTagType.Items.Add(New ListItem("Image Selector", "5"))
        ddlMetaTagType.Items.Add(New ListItem("ListSummary Selector", "3"))
    End If
End Sub
```

3. Add a button click event.

```
Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
btnAdd.Click
```

4. Create the following objects.

```
        Dim metaDataApi As New Ektron.Cms.API.Metadata
        Dim metaItem As New Ektron.Cms.ContentMetaData
```

5. Assign the following metaItem properties.

```
        metaItem.TypeName = txtMetaName.Text
        metaItem.Editable = chkEditable.Checked
        metaItem.MetaDisplayEE = chkDisplayEE.Checked
        metaItem.TagType = ddlMetaTagType.SelectedItem.Value
```

6. Create a Try/Catch around the AddMetaDataType method. This allows you to catch any exceptions and helps when debugging.

```
        Try
```

```
                    metaDataApi.AddMetaDataType(metaItem)
```

7. Check to make sure a valid TypeID was returned. If so, display the a message and the ID of the newly created metadata definition.

```
                    If metaItem.TypeId <> 0 Then
                        lblResult.Text = "The above metadata type definition was created in
CMS400.NET with ID = " & metaItem.TypeId
                    End If
```

8. If there is an error, let the user know what happened.

```
                Catch ex As Exception
                   lblResult.Text = ex.Message
                End Try
        End Sub
```

Metadata.DeleteMetadataType Method

Deletes a metadata type from the system.

C#

```
public DeleteMetadataType(Long Id);
```

Visual Basic

```
Public Sub DeleteMetadataType(ByVal Id As Long)
```

Parameters

Parameters	Description
ByVal Id As Long	Metatype ID

GetContentMetadataList Method

This is the overview for the GetContentMetadataList method overload.

Overload List

	Name	Description
	Metadata.GetContentMetadataList (Long) (see page 717)	Loads all of the metadata for the given content.
	Metadata.GetContentMetadataList (Long, Boolean) (see page 719)	Loads all of the metadata for the given content.

Metadata.GetContentMetadataList Method (Long)

Loads all of the metadata for the given content.

C#

```
public CustomAttributeList GetContentMetadataList(Long ContentId);
```

Visual Basic

```
Public Function GetContentMetadataList(ByVal ContentId As Long) As CustomAttributeList
```

Parameters

Parameters	Description
ByVal ContentId As Long	The contents ID.

Returns

Ektron.Cms.CustomAttributeList

Example

The following example shows how to create a Web page from which all metadata for a specific content ID can be retrieved. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetContentMetadataList(ByVal ContentId as Long) method. This method uses the InternalAdmin to retrieve the metadata for the content ID. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label.
```
<asp:Label ID="Label1" runat="server" Text="Content ID: "></asp:Label>
```
2. Add a text box to enter the content's ID.
```
<asp:TextBox ID="txtContentId" runat="server" Width="63px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Content ID box.
```
<asp:RequiredFieldValidator ID="rfvContentId" runat="server"
ControlToValidate="txtContentId"
                ErrorMessage="Please enter content ID"></asp:RequiredFieldValidator>
```
4. Add two breaks for spacing purposes.
```
<br /><br />
```
5. Add a button to get the metadata. We will set the button click event in the code behind.
```
<asp:Button ID="btnGetMetaData" runat="server" Text="Get MetaData" Width="91px" />
```
6. Add two more breaks for spacing purposes.
```
<br /><br />
```
7. Add a label to display to display the metadata information.
```
<asp:Label ID="lblResult" runat="server" Width="604px" Height="113px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.
```
      Protected Sub btnGetMetaData_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGetMetaData.Click
```
2. Create the following objects for the Metadata (see page 714) API (see page 669), Custom Attribute List and each Custom Attribute item.
```
Dim metaDataApi As New Ektron.Cms.API.Metadata
Dim metaDataList As New Ektron.Cms.CustomAttributeList
Dim metaItem As New Ektron.Cms.CustomAttribute
```
3. Check to make sure the content ID is numeric.
```
If Not IsNumeric(txtContentId.Text) Then
    lblResult.Text = "Invalid Content ID"
    Exit Sub
End If
```
4. Create a Try/Catch around the GetContentMetadataList method. This allows you to catch any exceptions and helps when debugging.
```
Try
    metaDataList = metaDataApi.GetContentMetadataList(CInt(txtContentId.Text))
```
5. Verify that items are returned and create a table to display the returned information.
```
    If metaDataList.AttributeList.Length > 0 Then
        lblResult.Text = "<table> <th>Metaname</th> <th align=""left""">Value</th> "
```
6. Populate the table with metadata information and display it.
```
        For Each metaItem In metaDataList.AttributeList
            If metaItem.Value.ToString().Length > 0 Then
                lblResult.Text &= "<tr>"
                lblResult.Text &= "<td>" & metaItem.Name & "</td>"
                lblResult.Text &= "<td>" & metaItem.Value & "</td>"
                lblResult.Text &= "</tr>"
            End If
        Next
        lblResult.Text &= "</table>"
```

8. If no metadata exists, display that information.

```
            Else
                lblResult.Text = "No metadata definitions found"
                Exit Sub
            End If
```

9. Display a message if there is an error.

```
        Catch ex As Exception
            lblResult.Text = ex.Message
        End Try
    End Sub
```

Metadata.GetContentMetadataList Method (Long, Boolean)

Loads all of the metadata for the given content.

C#

```
public CustomAttributeList GetContentMetadataList(Long ContentId, Boolean preview);
```

Visual Basic

```
Public Function GetContentMetadataList(ByVal ContentId As Long, ByVal preview As Boolean) As
CustomAttributeList
```

Parameters

Parameters	Description
ByVal ContentId As Long	The contents ID.
ByVal preview As Boolean	True implies get metadata from staging content, else use published content.

Returns

Ektron.Cms.CustomAttributeList

Metadata.GetIntegratedSearchFolder Method

Loads all custom folder data.

C#

```
public IntegratedFolderData GetIntegratedSearchFolder(Long Id);
```

Visual Basic

```
Public Function GetIntegratedSearchFolder(ByVal Id As Long) As IntegratedFolderData
```

Parameters

Parameters	Description
ByVal Id As Long	The integrated search folder ID

Returns

Ektron.Cms.IntegratedSearchData

Metadata.GetIntegratedSearchFolders Method

Loads all the custom folders.

C#

```
public IntegratedFolderData[] GetIntegratedSearchFolders();
```

Visual Basic

```
Public Function GetIntegratedSearchFolders() As IntegratedFolderData()
```

Returns

Ektron.Cms.IntegratedSearchData()

Metadata.GetMetadataByFolder Method

Returns a collection of ContentMetadata assigned to a folder.

C#

```
public ContentMetaData[] GetMetadataByFolder(Long folderId);
```

Visual Basic

```
Public Function GetMetadataByFolder(ByVal folderId As Long) As ContentMetaData()
```

Parameters

Parameters	Description
ByVal folderId As Long	folder to retrieve metadata for

Returns

array of Ektron.Cms.ContentMetadata objects

Metadata.GetMetadataType Method

Loads all content metadata.

C#

```
public ContentMetaData GetMetadataType(Long Id);
```

Visual Basic

```
Public Function GetMetadataType(ByVal Id As Long) As ContentMetaData
```

Parameters

Parameters	Description
ByVal Id As Long	The metadata type ID

Returns

Ektron.Cms.ContentMetaData

Metadata.GetMetaDataTypes Method

Loads all the metadata types.

C#

```
public ContentMetaData[] GetMetaDataTypes(String OrderBy);
```

Visual Basic

```
Public Function GetMetaDataTypes(ByVal OrderBy As String) As ContentMetaData()
```

Parameters

Parameters	Description
ByVal OrderBy As String	Order by clause

Returns

Ektron.Cms.ContentMetaData()

Metadata.UpdateMetadataType Method

Updates the metadata type.

C#

```
public UpdateMetadataType(Collection pagedata);
```

Visual Basic

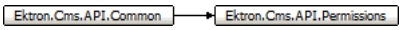
```
Public Sub UpdateMetadataType(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
ByVal pagedata As Collection	Metadata (see page 714) types

Permissions Class

Class Hierarchy



C#

```
public class Permissions : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Permissions
Inherits Ektron.Cms.API.Common
```

File

Permissions.vb

Description

This is class Ektron.Cms.API.Permissions.

Permissions Methods

The methods of the Permissions class are listed here.

Public Methods

	Name	Description
	AddItemPermission (see page 722)	Adds the item permissions for a user or a usergroup.
	DeleteItemPermission (see page 722)	Deletes a user or a group from the item's permissions
	DisableContentInheritance (see page 722)	Disable permissions inheritance on the content and copies current permissions to the content.
	DisableFolderInheritance (see page 723)	Disable permissions inheritance on the folder and copies current permissions to the folder.
	EnableContentInheritance (see page 723)	Deletes current permissions from the content and enable permissions inheritance on the content.
	EnableFolderInheritance (see page 723)	Deletes current permissions from the folder and enable permissions inheritance on the folder.
	GetUserPermissions (see page 723)	This is the overview for the GetUserPermissions method overload.
	LoadPermissions (see page 725)	Loads the permission for given object, such as content, folder, task, etc.
	UpdateItemPermission (see page 725)	Updates the item permissions for a user or a usergroup.

Permissions.AddItemPermission Method

Adds the item permissions for a user or a usergroup.

C#

```
public Boolean AddItemPermission(UserPermissionData userPermData);
```

Visual Basic

```
Public Function AddItemPermission(ByVal userPermData As UserPermissionData) As Boolean
```

Parameters

Parameters	Description
ByVal userPermData As UserPermissionData	User (see page 950)'s Permissions (see page 721) Data

Returns

False if it's successful

Permissions.DeleteItemPermission Method

Deletes a user or a group from the item's permissions

C#

```
public Boolean DeleteItemPermission(Long id, Boolean IsGroupId, Long itemId,
Ektron.Cms.Common.EkEnumeration.CMSObjectTypes itemType);
```

Visual Basic

```
Public Function DeleteItemPermission(ByVal id As Long, ByVal IsGroupId As Boolean, ByVal
itemId As Long, ByVal itemType As Ektron.Cms.Common.EkEnumeration.CMSObjectTypes) As Boolean
```

Parameters

Parameters	Description
ByVal id As Long	User (see page 950) or Group Id
ByVal IsGroupId As Boolean	Given ID is a user id or a group id?
ByVal itemId As Long	ContentId or FolderId
ByVal itemType As Ektron.Cms.Common.EkEnumeration.CMSObjectTypes	Given item id is EkEnumeration.CMSObjectTypes.Content (see page 822) or EkEnumeration.CMSObjectTypes.Folder (see page 683)?

Returns

False if it's successful.

Permissions.DisableContentInheritance Method

Disable permissions inheritance on the content and copies current permissions to the content.

C#

```
public Boolean DisableContentInheritance(Long id);
```

Visual Basic

```
Public Function DisableContentInheritance(ByVal id As Long) As Boolean
```

Parameters

Parameters	Description
ByVal id As Long	Content (see page 822) Id

Returns

False if it's successful.

Permissions.DisableFolderInheritance Method

Disable permissions inheritance on the folder and copies current permissions to the folder.

C#

```
public Boolean DisableFolderInheritance(Long id);
```

Visual Basic

```
Public Function DisableFolderInheritance(ByVal id As Long) As Boolean
```

Parameters

Parameters	Description
ByVal id As Long	folder Id

Returns

False if it's successful.

Permissions.EnableContentInheritance Method

Deletes current permissions from the content and enable permissions inheritance on the content.

C#

```
public Boolean EnableContentInheritance(Long id);
```

Visual Basic

```
Public Function EnableContentInheritance(ByVal id As Long) As Boolean
```

Parameters

Parameters	Description
ByVal id As Long	Content (see page 822) Id

Returns

False if it's successful.

Permissions.EnableFolderInheritance Method

Deletes current permissions from the folder and enable permissions inheritance on the folder.

C#

```
public Boolean EnableFolderInheritance(Long id);
```

Visual Basic

```
Public Function EnableFolderInheritance(ByVal id As Long) As Boolean
```

Parameters

Parameters	Description
ByVal id As Long	folder Id

Returns

False if it's successful.

GetUserPermissions Method

This is the overview for the GetUserPermissions method overload.

Overload List

	Name	Description
	Permissions.GetUserPermissions (Long, String, Long, String) (see page 724)	Loads the user permissions as a array of UserPermissionData.
	Permissions.GetUserPermissions (Long, String, Long, String, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) (see page 724)	Loads the user permissions as a array of UserPermissionData.

Permissions.GetUserPermissions Method (Long, String, Long, String)

Loads the user permissions as a array of UserPermissionData.

C#

```
public UserPermissionData[] GetUserPermissions(Long Id, String ItemType, Long UserId, String
UserGroupList);
```

Visual Basic

```
Public Function GetUserPermissions(ByVal Id As Long, ByVal ItemType As String, ByVal UserId As
Long, ByVal UserGroupList As String) As UserPermissionData()
```

Parameters

Parameters	Description
ByVal Id As Long	the item's ID
ByVal ItemType As String	The item's type
ByVal UserId As Long	The user's ID
ByVal UserGroupList As String	the group list

Returns

Ektron.Cms.UserPermissionData()

Permissions.GetUserPermissions Method (Long, String, Long, String, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType)

Loads the user permissions as a array of UserPermissionData.

C#

```
public UserPermissionData[] GetUserPermissions(Long Id, String ItemType, Long UserId, String
UserGroupList, ContentAPI.PermissionUserType PermissionType, ContentAPI.PermissionRequestType
PermissionRequest);
```

Visual Basic

```
Public Function GetUserPermissions(ByVal Id As Long, ByVal ItemType As String, ByVal UserId As
Long, ByVal UserGroupList As String, ByVal PermissionType As ContentAPI.PermissionUserType,
ByVal PermissionRequest As ContentAPI.PermissionRequestType) As UserPermissionData()
```

Parameters

Parameters	Description
ByVal Id As Long	the item's ID
ByVal ItemType As String	The item's type
ByVal UserId As Long	The user's ID
ByVal UserGroupList As String	the group list

ByVal PermissionType As ContentAPI.PermissionUserType	PermissionUserType
ByVal PermissionRequest As ContentAPI.PermissionRequestType	PermissionRequestType

Returns

Ektron.Cms.UserPermissionData()

Permissions.LoadPermissions Method

Loads the permission for given object, such as content, folder, task, etc.

C#

```
public PermissionData LoadPermissions(Long Id, String RequestType, Optional ByVal
ContentAPI.PermissionResultType Type);
```

Visual Basic

```
Public Function LoadPermissions(ByVal Id As Long, ByVal RequestType As String, Optional ByVal
Type As ContentAPI.PermissionResultType = contentAPI.PermissionResultType.Common) As
PermissionData
```

Parameters

Parameters	Description
ByVal Id As Long	The object ID
ByVal RequestType As String	The object type := content | folder | collections | tasks
Optional ByVal Type As ContentAPI.PermissionResultType = contentAPI.PermissionResultType.Common	The permission result type

Returns

Ektron.Cms.PermissionData

Remarks

User (see page 950) login with administrator privileges are required to perform this operation.

Permissions.UpdateItemPermission Method

Updates the item permissions for a user or a usergroup.

C#

```
public Boolean UpdateItemPermission(UserPermissionData userPermData);
```

Visual Basic

```
Public Function UpdateItemPermission(ByVal userPermData As UserPermissionData) As Boolean
```

Parameters

Parameters	Description
ByVal userPermData As UserPermissionData	User (see page 950)'s Permissions (see page 721) Data

Returns

False if it's successful

Site Class

Class Hierarchy

C#

```
public class Site : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Site
Inherits Ektron.Cms.API.Common
```

File

Site.vb

Description

This is class Ektron.Cms.API.Site.

Site Methods

The methods of the Site class are listed here.

Public Methods

	Name	Description
	BreakTaskPermissionInherit (see page 726)	Removes the task permissions for a given user.
	DeleteCookie (see page 727)	Removes the cookie from the site.
	GetAllActiveLanguages (see page 727)	Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language or Settings area.
	GetAllLanguages (see page 727)	This is the overview for the GetAllLanguages method overload.
	GetClientDayNames (see page 728)	Returns the day name of a given content language (or site language).
	GetClientMonthNames (see page 728)	Returns the month name of a given content language (or site language)
	GetCookieValue (see page 728)	Reads the cookie value.
	GetLanguage (see page 729)	Returns the details about the given language ID.
	GetLanguages (see page 729)	Gets a list of all languages that meet a given state, for example, active.
	GetSiteVariables (see page 729)	Returns all of the settings data.
	GetTaskPermission (see page 730)	Returns 1 if the user has one of the task permissions, otherwise 0.
	IsTaskPermissionInherited (see page 730)	Returns true if a user has already granted one of the task permissions.
	SetCookieValue (see page 730)	Resets the cookie value.
	SetTaskPermission (see page 731)	Updates the task permission(s) for the given user or group.
	UpdateSiteEnabledLanguages (see page 731)	Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled.
	UpdateSiteVariables (see page 731)	Update the site settings

Site.BreakTaskPermissionInherit Method

Removes the task permissions for a given user.

C#

```
public BreakTaskPermissionInherit(Long Id);
```

Visual Basic

```
Public Sub BreakTaskPermissionInherit(ByVal Id As Long)
```

Parameters

Parameters	Description
ByVal Id As Long	User (see page 950) ID

Site.DeleteCookie Method

Removes the cookie from the site.

C#

```
public DeleteCookie();
```

Visual Basic

```
Public Sub DeleteCookie()
```

Site.GetAllActiveLanguages Method

Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language or Settings area.

C#

```
public LanguageData[] GetAllActiveLanguages();
```

Visual Basic

```
Public Function GetAllActiveLanguages() As LanguageData()
```

Returns

Array of Ektron.Cms.LanguageData

GetAllLanguages Method

This is the overview for the GetAllLanguages method overload.

Overload List

	Name	Description
	Site.GetAllLanguages () (see page 727)	Gets a list of all the available languages sorted by language name in ascending order.
	Site.GetAllLanguages (LanguageDataSet.SortBy, LanguageDataSet.SortDirection) (see page 728)	Gets a list of all the available languages.

Site.GetAllLanguages Method ()

Gets a list of all the available languages sorted by language name in ascending order.

C#

```
public LanguageData[] GetAllLanguages();
```

Visual Basic

```
Public Function GetAllLanguages() As LanguageData()
```

Returns

Array of Ektron.Cms.LanguageData

Site.GetAllLanguages Method (LanguageDataSet.SortBy, LanguageDataSet.SortDirection)

Gets a list of all the available languages.

C#

```
public LanguageData[] GetAllLanguages(LanguageDataSet.SortBy orderBy,
LanguageDataSet.SortDirection direction);
```

Visual Basic

```
Public Function GetAllLanguages(ByVal orderBy As LanguageDataSet.SortBy, ByVal direction As
LanguageDataSet.SortDirection) As LanguageData()
```

Parameters

Parameters	Description
ByVal orderBy As LanguageDataSet.SortBy	Determines the sort order.
ByVal direction As LanguageDataSet.SortDirection	Determines the sort direction.

Returns

Array of Ektron.Cms.LanguageData

Site.GetClientDayNames Method

Returns the day name of a given content language (or site language).

C#

```
public String GetClientDayNames();
```

Visual Basic

```
Public Function GetClientDayNames() As String
```

Returns

String

Site.GetClientMonthNames Method

Returns the month name of a given content language (or site language)

C#

```
public String GetClientMonthNames();
```

Visual Basic

```
Public Function GetClientMonthNames() As String
```

Returns

String

Site.GetCookieValue Method

Reads the cookie value.

C#

```
public String GetCookieValue(String Keyname);
```

Visual Basic

```
Public Function GetCookieValue(ByVal Keyname As String) As String
```

Parameters

Parameters	Description
ByVal Keyname As String	The Keyname. (Required)

Returns

String

Site.GetLanguage Method

Returns the details about the given language ID.

C#

```
public LanguageData GetLanguage(Long Id);
```

Visual Basic

```
Public Function GetLanguage(ByVal Id As Long) As LanguageData
```

Parameters

Parameters	Description
ByVal Id As Long	Valid non negative language ID (refer to the language_type table for all the languages)

Returns

Ektron.Cms.LanguageData

Site.GetLanguages Method

Gets a list of all languages that meet a given state, for example, active.

C#

```
public LanguageDataSet.LanguageDataTable GetLanguages(LanguageDataSet.LanguageState State,
Optional ByVal LanguageDataSet.SortBy SortBy, Optional ByVal LanguageDataSet.SortDirection
SortDirection);
```

Visual Basic

```
Public Function GetLanguages(ByVal State As LanguageDataSet.LanguageState, Optional ByVal
SortBy As LanguageDataSet.SortBy = LanguageDataSet.SortBy.LanguageState, Optional ByVal
SortDirection As LanguageDataSet.SortDirection = LanguageDataSet.SortDirection.Ascending) As
LanguageDataSet.LanguageDataTable
```

Parameters

Parameters	Description
ByVal State As LanguageDataSet.LanguageState	The state that the language must meet. For example, defined, active, site enabled.
Optional ByVal SortBy As LanguageDataSet.SortBy = LanguageDataSet.SortBy.LanguageState	The field on which to sort the languages in the data table.
Optional ByVal SortDirection As LanguageDataSet.SortDirection = LanguageDataSet.SortDirection.Ascending	The direction, ascending or descending, in which to sort the field values.

Returns

LanguageDataTable of languages that meet the given state.

Site.GetSiteVariables Method

Returns all of the settings data.

C#

```
public SettingsData GetSiteVariables(Optional ByVal Long Id);
```

Visual Basic

```
Public Function GetSiteVariables(Optional ByVal Id As Long = -1) As SettingsData
```

Parameters

Parameters	Description
Optional ByVal Id As Long = -1	User (see page 950) ID

Returns

Ektron.Cms.SettingsData

Remarks

If user ID is passed it returns the additional value of MaxUsersAllowed, AdInstalled and AdAllowed properties.

Site.GetTaskPermission Method

Returns 1 if the user has one of the task permissions, otherwise 0.

C#

```
public Long GetTaskPermission(Long Id, int Type);
```

Visual Basic

```
Public Function GetTaskPermission(ByVal Id As Long, ByVal Type As Integer) As Long
```

Parameters

Parameters	Description
ByVal Id As Long	User (see page 950) or group ID
ByVal Type As Integer	User (see page 950) or group (1 = user, 2 = group).

Returns

Returns 1 if the user has one of the task permissions. Otherwise, it returns 0.

Site.IsTaskPermissionInherited Method

Returns true if a user has already granted one of the task permissions.

C#

```
public Boolean IsTaskPermissionInherited(Long Id);
```

Visual Basic

```
Public Function IsTaskPermissionInherited(ByVal Id As Long) As Boolean
```

Parameters

Parameters	Description
ByVal Id As Long	User (see page 950) ID

Returns

Boolean

Site.SetCookieValue Method

Resets the cookie value.

C#

```
public SetCookieValue(String Keyname, String CookieValue);
```

Visual Basic

```
Public Sub SetCookieValue(ByVal Keyname As String, ByVal CookieValue As String)
```

Parameters

Parameters	Description
ByVal Keyname As String	The Keyname. (Required)
ByVal CookieValue As String	Value

Site.SetTaskPermission Method

Updates the task permission(s) for the given user or group.

C#

```
public SetTaskPermission(Long Id, int UserType, int PermissionType);
```

Visual Basic

```
Public Sub SetTaskPermission(ByVal Id As Long, ByVal UserType As Integer, ByVal PermissionType
As Integer)
```

Parameters

Parameters	Description
ByVal Id As Long	User (see page 950) ID or Group ID
ByVal UserType As Integer	User (see page 950) (1) or Group (2)
ByVal PermissionType As Integer	Ektron.Cms.Common.TaskPermission (1 = Create Task (see page 909), 2 = Redirect Task (see page 909), 3 = Delete Task (see page 909), 4 = Edit Alias)

Site.UpdateSiteEnabledLanguages Method

Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled.

C#

```
public UpdateSiteEnabledLanguages(String LanguageIDs);
```

Visual Basic

```
Public Sub UpdateSiteEnabledLanguages(ByVal LanguageIDs As String)
```

Parameters

Parameters	Description
ByVal LanguageIDs As String	Comma-separated list of language IDs. For example, "1033,1031"

Site.UpdateSiteVariables Method

Update the site settings

C#

```
public UpdateSiteVariables(Hashtable item);
```

Visual Basic

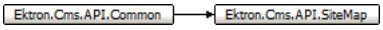
```
Public Sub UpdateSiteVariables(ByVal item As Hashtable)
```

Parameters

Parameters	Description
ByVal item As Hashtable	Set of settings data as key, value pair.

SiteMap Class

Class Hierarchy



C#

```
public class SiteMap : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class SiteMap
Inherits Ektron.Cms.API.Common
```

File

SiteMap.vb

Description

This is class Ektron.Cms.API.SiteMap.

SiteMap Methods

The methods of the SiteMap class are listed here.

Public Methods

	Name	Description
	GetFolderBreadcrumbPath (see page 732)	Get breadcrumb path of the folder in which content block exists.
	GetWebsiteMap (see page 734)	Creates the site map from folder breadcrumb.
	RenderFolderBreadcrumbHtml (see page 735)	Render folder breadcrumb HTML.
	RenderWebsiteMap (see page 736)	Create HTML to render on the page.

SiteMap.GetFolderBreadcrumbPath Method

Get breadcrumb path of the folder in which content block exists.

C#

```
public Ektron.Cms.Common.SitemapPath[] GetFolderBreadcrumbPath(Long id, Optional ByVal Boolean
isFolder);
```

Visual Basic

```
Public Function GetFolderBreadcrumbPath(ByVal id As Long, Optional ByVal isFolder As Boolean =
False) As Ektron.Cms.Common.SitemapPath()
```

Parameters

Parameters	Description
ByVal id As Long	Content (see page 822) or Folder (see page 683) ID.
Optional ByVal isFolder As Boolean = False	Set to True if the ID is a folder ID.

Returns

Array of nodes in the path.

Example

The following example shows how to create a Web page from which the breadcrumb path for a given folder can be retrieved. In this This example uses some standard drag and drop controls and a small section of VB code utilizing the

GetFolderBreadcrumbPath method. The GetFolderBreadcrumbPath method allows you to use a folder ID or a content's ID to retrieve the information. This method uses the InternalAdmin to retrieve the breadcrumb path. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a radio button list. We will populate the radio buttons in the code behind.
```
<asp:RadioButtonList ID="rblItem" runat="server" Height="64px"></asp:RadioButtonList>
```
2. Add two breaks for spacing purposes.
```
<br /><br />
```
3. Add a label to show where the ID goes.
```
<asp:Label ID="Label6" runat="server" Text="ID:" Width="76px"></asp:Label>
```
4. Add a text box for the ID.
```
<asp:TextBox ID="txtId" runat="server" Width="55px"></asp:TextBox>
```
5. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the ID box.
```
<asp:RequiredFieldValidator ID="rfvId" runat="server" ControlToValidate="txtId"
   ErrorMessage="Please enter a content/folder ID"
Width="179px"></asp:RequiredFieldValidator>
```
6. Add two more breaks for spacing purposes.
```
<br /><br />
```
7. Add a button to submit the ID to be queried. We will set the button click event in the code behind.
```
<asp:Button ID="btnGet" runat="server" Text="Get Breadcrumb" Width="105px" />
```
8. Add two last breaks for spacing purposes.
```
<br /><br />
```
9. Add a label to display the breadcrumb path.
```
<asp:Label ID="lblResult" runat="server" Width="502px" Height="100px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. In the page load event, populate the radio buttons.
```
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
      If Not IsPostBack() Then
          rblItem.Items.Add(New ListItem("Folder", "true"))
          rblItem.Items.Add(New ListItem("Content", "false"))
          rblItem.Items(0).Selected = True
      End If
    End Sub
```
2. Add a button click event.
```
    Protected Sub btnGet_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
btnGet.Click
```
3. Check to make sure the ID is numeric.
```
        If Not IsNumeric(txtId.Text) Then
            lblResult.Text = "Invalid content/folder ID"
            Exit Sub
        End If
```
4. Create an object for the Site (see page 725) Map (see page 350) API (see page 669).
```
        Dim sitemapApi As New Ektron.Cms.API.SiteMap
```
5. Create an object for the site map data.
```
        Dim mapData As Ektron.Cms.Common.SitemapPath()
```
6. Create a Try/Catch around the GetFolderBreadcrumbPath method. This allows you to catch any exceptions and helps when debugging.
```
        Try
            mapData = sitemapApi.GetFolderBreadcrumbPath(CInt(txtId.Text),
CBool(rblItem.SelectedValue))
```
7. Check to make sure map data is returned. If map data is returned, display the information.
```
            If (Not mapData Is Nothing) AndAlso (mapData.Length > 0) Then
                lblResult.Text = sitemapApi.RenderFolderBreadcrumbHtml(mapData, ">", True,
```

```
"", False, "")
```

8. If no map data exists, display a message.

```
                Else
                    lblResult.Text = "No breadcrumb path found"
                End If
```

9. Display a message if there is an error.

```
               Catch ex As Exception
                  lblResult.Text = ex.Message
               End Try
        End Sub
```

SiteMap.GetWebsiteMap Method

Creates the site map from folder breadcrumb.

C#

```
public WebsiteMap GetWebsiteMap(Optional ByVal Long folderId, Optional ByVal Boolean
getSameLevelNodes);
```

Visual Basic

```
Public Function GetWebsiteMap(Optional ByVal folderId As Long = 0, Optional ByVal
getSameLevelNodes As Boolean = False) As WebsiteMap
```

Parameters

Parameters	Description
Optional ByVal folderId As Long = 0	Build the web sitemap from the folder specified.
Optional ByVal getSameLevelNodes As Boolean = False	Do you want to retrieve all nodes at the same level as the one you specify? **True** = include all sites at the same level as the `folderId` property. **False** = Do not include the same level nodes.

Example

The following example shows how to create a Web page that displays the site map for a given folder. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetWebsiteMap method. This method uses the InternalAdmin to access and return the site map information. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label:
```
<asp:Label ID="Label6" runat="server" Text="Folder ID:" Width="66px"></asp:Label>
```
2. Add text box where the folder ID can be entered.
```
<asp:TextBox ID="txtFolderId" runat="server" Width="55px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder ID box.
```
<asp:RequiredFieldValidator ID="rfvId" runat="server" ControlToValidate="txtFolderId"
   ErrorMessage="Please enter folder ID" Width="179px"></asp:RequiredFieldValidator>
```
4. Add two breaks for spacing purposes.
```
<br /><br />
```
5. Add a button to submit the folder ID and retrieve the site map information. We will set the button click event in the code behind.
```
<asp:Button ID="btnGet" runat="server" Text="Get site map" Width="91px" />
```
6. Add two more breaks for spacing purposes.
```
<br /><br />
```
7. Add a label to display the site map.
```
<asp:Label ID="lblResult" runat="server" Width="502px" Height="100px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
       Protected Sub btnGet_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
btnGet.Click
```

2. Check to make sure the folder ID is numeric.

```
If Not IsNumeric(txtFolderId.Text) Then
    lblResult.Text = "Invalid Folder ID"
    Exit Sub
End If
```

3. Create an object for the site map API (see page 669).

```
Dim sitemapApi As New Ektron.Cms.API.SiteMap
```

4. Create an object for the map data.

```
Dim mapData As Ektron.Cms.Common.WebsiteMap
```

5. Create a Try/Catch around the GetWebsiteMap method. This allows you to catch any exceptions and helps when debugging.

```
Try
    mapData = sitemapApi.GetWebsiteMap(CInt(txtFolderId.Text), False)
```

6. Check to make sure child nodes are returned. If child nodes are returned, display the information.

```
    If (Not mapData.childrenNodes Is Nothing) Then
        lblResult.Text = sitemapApi.RenderWebsiteMap(mapData, "")
```

7. If no child nodes are returned, display a message.

```
    Else
        lblResult.Text = "No breadcrumb path found"
    End If
```

8. Display a message if there is an error.

```
    Catch ex As Exception
       lblResult.Text = ex.Message
    End Try
End Sub
```

SiteMap.RenderFolderBreadcrumbHtml Method

Render folder breadcrumb HTML.

C#

```
public String RenderFolderBreadcrumbHtml(Ektron.Cms.Common.SitemapPath nodes, String
pathSeparator, Boolean linkNodes, String linkTarget, Boolean displayVertical, String pageUrl);
```

Visual Basic

```
Public Function RenderFolderBreadcrumbHtml(ByVal nodes() As Ektron.Cms.Common.SitemapPath,
ByVal pathSeparator As String, ByVal linkNodes As Boolean, ByVal linkTarget As String, ByVal
displayVertical As Boolean, ByVal pageUrl As String) As String
```

Parameters

Parameters	Description
ByVal nodes() As Ektron.Cms.Common.SitemapPath	The items contained in the breadcrumb path. Ektron.Cms.Common.SitemapPath
ByVal pathSeparator As String	a special character that separates the items in a breadcrumb trail.
ByVal linkNodes As Boolean	Should the items in the breadcrumb path be Hyperlinked? **True** = hyperlink breadcrumb items. **False** = breadcrumb items appear as plain text.
ByVal linkTarget As String	Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self. • **_Self** - opens link in the same window. • **_Top** - opens link in a parent window. • **_Blank** - opens link in a new window. • **_Parent** - opens link in the parent frame.

ByVal displayVertical As Boolean	Do you want the breadcrumb path to display vertically? **True** = display the breadcrumb trail vertically. **False** = display the breadcrumb trail horizontally.
ByVal pageUrl As String	The current page URL. Usually Request.Url.

Returns

HTML

SiteMap.RenderWebsiteMap Method

Create HTML to render on the page.

C#

```
public String RenderWebsiteMap(WebsiteMap webSitemap, String displayType, Optional ByVal
String className, Optional ByVal int maxLevel, Optional ByVal int startLevel, Optional ByVal
String PageUrl);
```

Visual Basic

```
Public Function RenderWebsiteMap(ByVal webSitemap As WebsiteMap, ByVal displayType As String,
Optional ByVal className As String = "ektSitemap", Optional ByVal maxLevel As Integer = 0,
Optional ByVal startLevel As Integer = 0, Optional ByVal PageUrl As String = "") As String
```

Parameters

Parameters	Description
ByVal webSitemap As WebsiteMap	GetWebSiteMap()
ByVal displayType As String	Determines how to produce HTML.
Optional ByVal className As String = "ektSitemap"	Stylesheet class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.
Optional ByVal maxLevel As Integer = 0	Set the max level of the sitemap. Set it to 0 for unlimited.
Optional ByVal startLevel As Integer = 0	Set the starting level of the sitemap. Set it to 0 for start from root.

Returns

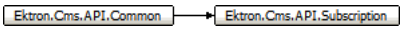
HTML/XML

Remarks

Supported displayType: ListWithDescription | List | RawXml

Subscription Class

Class Hierarchy



C#

```
public class Subscription : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Subscription
Inherits Ektron.Cms.API.Common
```

File

Subscription.vb

Description

This is class Ektron.Cms.API.Subscription.

Subscription Methods

The methods of the Subscription class are listed here.

Public Methods

	Name	Description
	AddEmailFrom (see page 737)	Adds a subscription email address.
	AddSubscription (see page 738)	Adds a subscription.
	AddSubscriptionMessage (see page 738)	Adds subscription message(s).
	DeleteEmailFrom (see page 738)	Deletes a subscription email address from system.
	DeleteSubscription (see page 738)	Deletes a subscription from the system.
	DeleteSubscriptionMessage (see page 739)	Deletes the subscription message from the system.
	GetAllActiveSubscriptions (see page 739)	Loads all of the active subscriptions.
	GetAllEmailFrom (see page 739)	Returns all of the email as an array of EmailFromData.
	GetAllSubscriptionMessages (see page 739)	Loads all the subscription messages.
	GetAllSubscriptions (see page 740)	Loads all the subscriptions.
	GetFolderInheritedFrom (see page 740)	Retrieves the inherited parent ID.
	GetSubscription (see page 740)	Retrieves a subscription by using its ID.
	GetSubscriptionMessage (see page 740)	Retrieves a subscription message by using the subscription ID.
	GetSubscriptionMessagesForType (see page 741)	Retrieve subscription message for type.
	GetSubscriptionPropertiesForContent (see page 741)	Retrieves the subscription properties for content using the content's ID.
	GetSubscriptionPropertiesForFolder (see page 741)	Retrieves the subscription properties for a given folder.
	GetSubscriptionsForContent (see page 742)	Loads the subscriptions for the given content.
	GetSubscriptionsForFolder (see page 742)	Loads the subscriptions for a given folder.
	SendWebAlert (see page 742)	Triggers the web alert subsystem. Requires that the subscription properties be set for the content first.
	UpdateEmailFrom (see page 743)	Update the email address.
	UpdateSubscription (see page 743)	Updates the subscription.
	UpdateSubscriptionMessage (see page 743)	Updates the subscription message.
	UpdateSubscriptionPropertiesForContent (see page 743)	Sets the subscription properties for a single piece of content.
	UpdateSubscriptionPropertiesForFolder (see page 744)	Sets the subscription properties for a single folder.

Subscription.AddEmailFrom Method

Adds a subscription email address.

C#

```
public AddEmailFrom(Collection data);
```

Visual Basic

```
Public Sub AddEmailFrom(ByVal data As Collection)
```

Parameters

Parameters	Description
ByVal data As Collection	Email address as a collection.

Subscription.AddSubscription Method

Adds a subscription.

C#

```
public AddSubscription(Collection data);
```

Visual Basic

```
Public Sub AddSubscription(ByVal data As Collection)
```

Parameters

Parameters	Description
ByVal data As Collection	The subscription's name.

Subscription.AddSubscriptionMessage Method

Adds subscription message(s).

C#

```
public AddSubscriptionMessage(Collection data);
```

Visual Basic

```
Public Sub AddSubscriptionMessage(ByVal data As Collection)
```

Parameters

Parameters	Description
ByVal data As Collection	Subscription (see page 736) message data.

Subscription.DeleteEmailFrom Method

Deletes a subscription email address from system.

C#

```
public DeleteEmailFrom(Collection data);
```

Visual Basic

```
Public Sub DeleteEmailFrom(ByVal data As Collection)
```

Parameters

Parameters	Description
ByVal data As Collection	Email data.

Subscription.DeleteSubscription Method

Deletes a subscription from the system.

C#

```
public DeleteSubscription(Collection data);
```

Visual Basic

```
Public Sub DeleteSubscription(ByVal data As Collection)
```

Parameters

Parameters	Description
ByVal data As Collection	Subscription (see page 736) data

Subscription.DeleteSubscriptionMessage Method

Deletes the subscription message from the system.

C#

```
public DeleteSubscriptionMessage(Collection data);
```

Visual Basic

```
Public Sub DeleteSubscriptionMessage(ByVal data As Collection)
```

Parameters

Parameters	Description
ByVal data As Collection	Subscription (see page 736) message data

Subscription.GetAllActiveSubscriptions Method

Loads all of the active subscriptions.

C#

```
public SubscriptionData[] GetAllActiveSubscriptions();
```

Visual Basic

```
Public Function GetAllActiveSubscriptions() As SubscriptionData()
```

Returns

Ektron.Cms.SubscriptionData()

Subscription.GetAllEmailFrom Method

Returns all of the email as an array of EmailFromData.

C#

```
public EmailFromData[] GetAllEmailFrom();
```

Visual Basic

```
Public Function GetAllEmailFrom() As EmailFromData()
```

Returns

Ektron.Cms.EmailFromData()

Subscription.GetAllSubscriptionMessages Method

Loads all the subscription messages.

C#

```
public EmailMessageData[] GetAllSubscriptionMessages();
```

Visual Basic

```
Public Function GetAllSubscriptionMessages() As EmailMessageData()
```

Returns

Ektron.Cms.EmailMessageData()

Subscription.GetAllSubscriptions Method

Loads all the subscriptions.

C#

```
public SubscriptionData[] GetAllSubscriptions();
```

Visual Basic

```
Public Function GetAllSubscriptions() As SubscriptionData()
```

Returns

Ektron.Cms.SubscriptionData()

Subscription.GetFolderInheritedFrom Method

Retrieves the inherited parent ID.

C#

```
public Long GetFolderInheritedFrom(Long FolderId);
```

Visual Basic

```
Public Function GetFolderInheritedFrom(ByVal FolderId As Long) As Long
```

Parameters

Parameters	Description
ByVal FolderId As Long	The current folder's numeric ID.

Returns

The numeric ID of parent folder from which the folder is inherited.

Subscription.GetSubscription Method

Retrieves a subscription by using its ID.

C#

```
public SubscriptionData GetSubscription(Long Id);
```

Visual Basic

```
Public Function GetSubscription(ByVal Id As Long) As SubscriptionData
```

Parameters

Parameters	Description
ByVal Id As Long	The subscription's ID.

Returns

Ektron.Cms.SubscriptionData

Subscription.GetSubscriptionMessage Method

Retrieves a subscription message by using the subscription ID.

C#

```
public EmailMessageData GetSubscriptionMessage(Long Id);
```

Visual Basic

```
Public Function GetSubscriptionMessage(ByVal Id As Long) As EmailMessageData
```

Parameters

Parameters	Description
ByVal Id As Long	The subscription's ID.

Returns

Ektron.Cms.EmailMessageData

Subscription.GetSubscriptionMessagesForType Method

Retrieve subscription message for type.

C#

```
public EmailMessageData[]
GetSubscriptionMessagesForType(Ektron.Cms.Common.EkEnumeration.EmailMessageTypes MessageType);
```

Visual Basic

```
Public Function GetSubscriptionMessagesForType(ByVal MessageType As
Ektron.Cms.Common.EkEnumeration.EmailMessageTypes) As EmailMessageData()
```

Parameters

Parameters	Description
ByVal MessageType As Ektron.Cms.Common.EkEnumeration.EmailMessageTypes	The message type.

Returns

Ektron.Cms.EmailMessageData()

Subscription.GetSubscriptionPropertiesForContent Method

Retrieves the subscription properties for content using the content's ID.

C#

```
public SubscriptionPropertiesData GetSubscriptionPropertiesForContent(Long ContentId);
```

Visual Basic

```
Public Function GetSubscriptionPropertiesForContent(ByVal ContentId As Long) As
SubscriptionPropertiesData
```

Parameters

Parameters	Description
ByVal ContentId As Long	The content's ID

Returns

Ektron.Cms.SubscriptionPropertiesData

Subscription.GetSubscriptionPropertiesForFolder Method

Retrieves the subscription properties for a given folder.

C#

```
public SubscriptionPropertiesData GetSubscriptionPropertiesForFolder(Long FolderId);
```

Visual Basic

```
Public Function GetSubscriptionPropertiesForFolder(ByVal FolderId As Long) As
SubscriptionPropertiesData
```

Parameters

Parameters	Description
ByVal FolderId As Long	The folder's ID.

Returns

Ektron.Cms.SubscriptionPropertiesData

Subscription.GetSubscriptionsForContent Method

Loads the subscriptions for the given content.

C#

```
public SubscriptionData[] GetSubscriptionsForContent(Long ContentId);
```

Visual Basic

```
Public Function GetSubscriptionsForContent(ByVal ContentId As Long) As SubscriptionData()
```

Parameters

Parameters	Description
ByVal ContentId As Long	The content's ID.

Returns

Ektron.Cms.SubscriptionData()

Subscription.GetSubscriptionsForFolder Method

Loads the subscriptions for a given folder.

C#

```
public SubscriptionData[] GetSubscriptionsForFolder(Long FolderId);
```

Visual Basic

```
Public Function GetSubscriptionsForFolder(ByVal FolderId As Long) As SubscriptionData()
```

Parameters

Parameters	Description
ByVal FolderId As Long	The folder's ID.

Returns

Ektron.Cms.SubscriptionData()

Subscription.SendWebAlert Method

Triggers the web alert subsystem. Requires that the subscription properties be set for the content first.

C#

```
public Boolean SendWebAlert(Long contentId);
```

Visual Basic

```
Public Function SendWebAlert(ByVal contentId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal contentId As Long	The ID of the content to set the subscription properties on.

Returns

Boolean.

Subscription.UpdateEmailFrom Method

Update the email address.

C#

```
public UpdateEmailFrom(Collection data);
```

Visual Basic

```
Public Sub UpdateEmailFrom(ByVal data As Collection)
```

Parameters

Parameters	Description
ByVal data As Collection	email data

Subscription.UpdateSubscription Method

Updates the subscription.

C#

```
public UpdateSubscription(Collection data);
```

Visual Basic

```
Public Sub UpdateSubscription(ByVal data As Collection)
```

Parameters

Parameters	Description
ByVal data As Collection	The subscription details.

Subscription.UpdateSubscriptionMessage Method

Updates the subscription message.

C#

```
public UpdateSubscriptionMessage(Collection data);
```

Visual Basic

```
Public Sub UpdateSubscriptionMessage(ByVal data As Collection)
```

Parameters

Parameters	Description
ByVal data As Collection	The subscription message details.

Subscription.UpdateSubscriptionPropertiesForContent Method

Sets the subscription properties for a single piece of content.

C#

```
public Boolean UpdateSubscriptionPropertiesForContent(Long contentId,
SubscriptionPropertiesData subscriptionProperties);
```

Visual Basic

```
Public Function UpdateSubscriptionPropertiesForContent(ByVal contentId As Long, ByVal
subscriptionProperties As SubscriptionPropertiesData) As Boolean
```

Parameters

Parameters	Description
ByVal contentId As Long	The ID of the content to set the subscription properties on.
ByVal subscriptionProperties As SubscriptionPropertiesData	The subscription properties class.

Returns

Boolean.

Subscription.UpdateSubscriptionPropertiesForFolder Method

Sets the subscription properties for a single folder.

C#

```
public Boolean UpdateSubscriptionPropertiesForFolder(Long folderId, SubscriptionPropertiesData
subscriptionProperties);
```

Visual Basic

```
Public Function UpdateSubscriptionPropertiesForFolder(ByVal folderId As Long, ByVal
subscriptionProperties As SubscriptionPropertiesData) As Boolean
```

Parameters

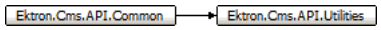
Parameters	Description
ByVal folderId As Long	The ID of the folder to set the subscription properties on.
ByVal subscriptionProperties As SubscriptionPropertiesData	The subscription properties class.

Returns

Boolean.

Utilities Class

Class Hierarchy



C#

```
public class Utilities : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Utilities
Inherits Ektron.Cms.API.Common
```

File

Utilities.vb

Description

This is class Ektron.Cms.API.Utilities.

Utilities Methods

The methods of the Utilities class are listed here.

Public Methods

	Name	Description
	LoadPermissions (see page 745)	Loads the permission for given object, such as content, folder, task, etc.

Utilities.LoadPermissions Method

Loads the permission for given object, such as content, folder, task, etc.

C#

```
[Obsolete("Use LoadPermissions from the Ektron.CMS.API.Permissions namespace.")]
public PermissionData LoadPermissions(Long Id, String RequestType, Optional ByVal
PermissionResultType Type);
```

Visual Basic

```
<Obsolete("Use LoadPermissions from the Ektron.CMS.API.Permissions namespace.")>
Public Function LoadPermissions(ByVal Id As Long, ByVal RequestType As String, Optional ByVal
Type As PermissionResultType = PermissionResultType.Common) As PermissionData
```

Parameters

Parameters	Description
ByVal Id As Long	The object ID
ByVal RequestType As String	The object type := content | folder | collections | tasks
Optional ByVal Type As PermissionResultType = PermissionResultType.Common	The permission result type

Returns

Ektron.Cms.PermissionData

Remarks

Use LoadPermissions from the Ektron.CMS.API.Permissions (see page 721) namespace.

Ektron.Cms.API.Analytics Namespace

This is namespace Ektron.Cms.API.Analytics.

Classes

	Name	Description
	Analytics (see page 745)	A class for managing and querying CMS400.NET site activity information

Classes

The following table lists classes in this documentation.

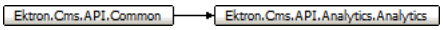
Classes

	Name	Description
	Analytics (see page 745)	A class for managing and querying CMS400.NET site activity information

Analytics Class

A class for managing and querying CMS400.NET site activity information

Class Hierarchy



C#

```
public class Analytics : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Analytics
Inherits Ektron.Cms.API.Common
```

File

Analytics.vb

Analytics Methods

The methods of the Analytics class are listed here.

Public Methods

	Name	Description
	GetSessionRank (see page 746)	Gets the Session rank of the visiting user. A session rank of 0 is the first page a new user has visited. A session rank of 1 is the first page a returning user has visited. A session rank of 2 is any other page.
	HardDumpAnalyticsBuffer (see page 746)	Forces the analytics buffer to write to the database. This surpasses the checks to see if the time before write or number of records have passed.
	QueryAnalytics (see page 746)	Provides a simple way of querying the database for analytics data. Returns the result as a weakly typed DataSet. Note that the connection is to the Analytics (see page 745) database and not to the standard CMS database.
	SoftDumpAnalyticsBuffer (see page 747)	Conditionally writes the analytics buffer in memory to write to the database. If the time before write or number of records before write conditions have been met or exceeded, this function will force the write to the database.
	TrackVisit (see page 747)	This is the overview for the TrackVisit method overload.

Analytics.GetSessionRank Method

Gets the Session rank of the visiting user. A session rank of 0 is the first page a new user has visited. A session rank of 1 is the first page a returning user has visited. A session rank of 2 is any other page.

C#

```
public int GetSessionRank(System.Web.UI.Page page);
```

Visual Basic

```
Public Function GetSessionRank(ByVal page As System.Web.UI.Page) As Integer
```

Parameters

Parameters	Description
ByVal page As System.Web.UI.Page	

Analytics.HardDumpAnalyticsBuffer Method

Forces the analytics buffer to write to the database. This surpasses the checks to see if the time before write or number of records have passed.

C#

```
public HardDumpAnalyticsBuffer();
```

Visual Basic

```
Public Sub HardDumpAnalyticsBuffer()
```

Analytics.QueryAnalytics Method

Provides a simple way of querying the database for analytics data. Returns the result as a weakly typed DataSet. Note that the connection is to the Analytics (see page 745) database and not to the standard CMS database.

C#

```
public DataSet QueryAnalytics(String selectCommands);
```

Visual Basic

```
Public Function QueryAnalytics(ByVal selectCommands() As String) As DataSet
```

Parameters

Parameters	Description
ByVal selectCommands() As String	An array of SQL queries in string format.

Returns

A weakly typed dataset with the result of the query.

Analytics.SoftDumpAnalyticsBuffer Method

Conditionally writes the analytics buffer in memory to write to the database. If the time before write or number of records before write conditions have been met or exceeded, this function will force the write to the database.

C#

```
public Boolean SoftDumpAnalyticsBuffer();
```

Visual Basic

```
Public Function SoftDumpAnalyticsBuffer() As Boolean
```

TrackVisit Method

This is the overview for the TrackVisit method overload.

Description

Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer. This method is overloaded.

Overload List

	Name	Description
	Analytics.TrackVisit (Long, Long, String, Integer, String, String) (see page 747)	Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer (see page 747).
	Analytics.TrackVisit (System.Web.UI.Page, Long) (see page 748)	Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer (see page 747).

Analytics.TrackVisit Method (Long, Long, String, Integer, String, String)

Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer (see page 747).

C#

```
public TrackVisit(Long content_id, Long user_id, String visitor_id, int session_rank, String
referring_url, String url);
```

Visual Basic

```
Public Sub TrackVisit(ByVal content_id As Long, ByVal user_id As Long, ByVal visitor_id As
String, ByVal session_rank As Integer, ByVal referring_url As String, ByVal url As String)
```

Parameters

Parameters	Description
ByVal content_id As Long	The ID of a content block that a visitor has viewed.
ByVal user_id As Long	The CMS user ID of either a CMS User (see page 950) or Member. Is zero for a visitor.
ByVal visitor_id As String	A unique ID used for tracking purposes.
ByVal session_rank As Integer	The rank of the session of a particular visitor. A session rank of 0 is the first page a new user has visited. A session rank of 1 is the first page a returning user has visited. A session rank of 2 is any other page.
ByVal referring_url As String	An external URL a user has arrived from.
ByVal url As String	The URL the user is visiting.

Analytics.TrackVisit Method (System.Web.UI.Page, Long)

Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer (see page 747).

C#

```
public TrackVisit(System.Web.UI.Page page, Long content_id);
```

Visual Basic

```
Public Sub TrackVisit(ByVal page As System.Web.UI.Page, ByVal content_id As Long)
```

Parameters

Parameters	Description
ByVal page As System.Web.UI.Page	The System.Web.UI.Page object of a page visit. Will extract information from it such as visitor id, user id, session rank, and so forth.
ByVal content_id As Long	The ID of a content block that a visitor has viewed.

Analytics Properties

The properties of the Analytics class are listed here.

Public Properties

	Name	Description
R	AnalyticsEnabled (see page 748)	Indicates if Analytics (see page 745) has been enabled through a setting in the web.config file. If disabled, no analytics tracking will be done. If this property is set to true, tracking on the control level can still be disabled through the "EnableAnalytics" property in the AnalyticsTracker (see page 25) server control.
R	AppendToIISLog (see page 749)	Indicates whether or not the web.config has been set to append user_id and content_id information to the IIS log.
R	InMemoryContentHitsData (see page 749)	The dataset that stores all the page/content view information in memory until it reaches its record or time threshold and writes back to the database. This is a strongly typed dataset of type 'ContentHitsData'. The underlying static member is initialized on demand.
R	RecordsBeforeWrite (see page 749)	Indicates the number of records to hold in the static memory buffer before writing to the database. This is used mainly to smooth out the number of writes to the database. This value and the 'TimeBeforeWrite (see page 749)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.
R	TimeBeforeWrite (see page 749)	Indicates the number of seconds before writing the in-memory buffer of records to the database. This is used mainly to smooth out the number of writes to the database. This value and the 'RecordsBeforeWrite (see page 749)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.

Analytics.AnalyticsEnabled Property

Indicates if Analytics (see page 745) has been enabled through a setting in the web.config file. If disabled, no analytics tracking will be done. If this property is set to true, tracking on the control level can still be disabled through the "EnableAnalytics" property in the AnalyticsTracker (see page 25) server control.

C#

```
public ReadOnly Boolean AnalyticsEnabled;
```

Visual Basic

```
Public ReadOnly Property AnalyticsEnabled() As Boolean
```

Description

True indicates that analytics data is being tracked. False indicates that no analytics data is being tracked.

Analytics.AppendToIISLog Property

Indicates whether or not the web.config has been set to append user_id and content_id information to the IIS log.

C#

```
public ReadOnly Boolean AppendToIISLog;
```

Visual Basic

```
Public ReadOnly Property AppendToIISLog() As Boolean
```

Description

Boolean

Analytics.InMemoryContentHitsData Property

The dataset that stores all the page/content view information in memory until it reaches its record or time threshold and writes back to the database. This is a strongly typed dataset of type 'ContentHitsData'. The underlying static member is initialized on demand.

C#

```
public ReadOnly ContentHitsData InMemoryContentHitsData;
```

Visual Basic

```
Public ReadOnly Property InMemoryContentHitsData() As ContentHitsData
```

Analytics.RecordsBeforeWrite Property

Indicates the number of records to hold in the static memory buffer before writing to the database. This is used mainly to smooth out the number of writes to the database. This value and the 'TimeBeforeWrite (see page 749)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.

C#

```
public ReadOnly int RecordsBeforeWrite;
```

Visual Basic

```
Public ReadOnly Property RecordsBeforeWrite() As Integer
```

Analytics.TimeBeforeWrite Property

Indicates the number of seconds before writing the in-memory buffer of records to the database. This is used mainly to smooth out the number of writes to the database. This value and the 'RecordsBeforeWrite (see page 749)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.

C#

```
public ReadOnly int TimeBeforeWrite;
```

Visual Basic

```
Public ReadOnly Property TimeBeforeWrite() As Integer
```

Ektron.Cms.API.Community Namespace

The Community API (see page 669) contains classes that allow you to manage aspects of your community site.

Classes

	Name	Description
	CommunityGroup (see page 750)	The CommunityGroup class allows you to work with community groups in CMS400.NET. For example, you can add a community group (AddCommunityGroup (see page 752)), get a list a list of community groups (GetAllCommunityGroups (see page 754)) or add a user to a community group (AddUserToCommunityGroup (see page 753)).
	Favorites (see page 763)	The Favorites class allows you to work with a user's favorites. Favorites are content items that a user has designated as their favorite content. From this class, you can add content to a user's favorites (AddContentFavorite (see page 764)), remove content from a user's favorites (DeleteContentFavorite (see page 765)) or get a list of favorites for a given user (GetContentFavorites (see page 768)).
	Flagging (see page 771)	The Flagging class allows you to work with Flag (see page 1231) Objects, Flag (see page 1231) Entries and content that has been flagged. For example, you can assign a flag to content (AssignFlagToContent (see page 772)) or get a list of flags for a content item (GetContentFlagData (see page 774)).
	Friends (see page 777)	Friends, which are the same as Colleagues in CMS400.NET, allow users to associate themselves with other users in the community. This API (see page 669) class allows you to manage colleagues for users in the system. For example, you can add a colleague (AddFriend (see page 779)), remove a colleague (DeleteFriend (see page 780)) or get a list of current colleagues (GetFriends (see page 782)).
	MessageBoard (see page 787)	The MessageBoard class allows you to work with message board entries associated with a user, community group or content item. For example, you can add a message to a message board (AddMessageBoardEntry (see page 788)), delete a message (DeleteMessageBoardEntry (see page 794)) or get a list of message board items for an object (GetAllMessageBoardEntries (see page 796)).
	Tags (see page 805)	The Tags class allows you to manage tags for content, users, and community groups. For example, you can add a tag to the system (AddTag (see page 806)), add a tag to a user (AddTagToUser (see page 811)), get a list of tags associated with a user (GetTagsForUser (see page 820)) or delete a tag by its ID (DeleteTagById (see page 812)).

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	CommunityGroup (see page 750)	The CommunityGroup class allows you to work with community groups in CMS400.NET. For example, you can add a community group (AddCommunityGroup (see page 752)), get a list a list of community groups (GetAllCommunityGroups (see page 754)) or add a user to a community group (AddUserToCommunityGroup (see page 753)).
	Favorites (see page 763)	The Favorites class allows you to work with a user's favorites. Favorites are content items that a user has designated as their favorite content. From this class, you can add content to a user's favorites (AddContentFavorite (see page 764)), remove content from a user's favorites (DeleteContentFavorite (see page 765)) or get a list of favorites for a given user (GetContentFavorites (see page 768)).
	Flagging (see page 771)	The Flagging class allows you to work with Flag (see page 1231) Objects, Flag (see page 1231) Entries and content that has been flagged. For example, you can assign a flag to content (AssignFlagToContent (see page 772)) or get a list of flags for a content item (GetContentFlagData (see page 774)).
	Friends (see page 777)	Friends, which are the same as Colleagues in CMS400.NET, allow users to associate themselves with other users in the community. This API (see page 669) class allows you to manage colleagues for users in the system. For example, you can add a colleague (AddFriend (see page 779)), remove a colleague (DeleteFriend (see page 780)) or get a list of current colleagues (GetFriends (see page 782)).
	MessageBoard (see page 787)	The MessageBoard class allows you to work with message board entries associated with a user, community group or content item. For example, you can add a message to a message board (AddMessageBoardEntry (see page 788)), delete a message (DeleteMessageBoardEntry (see page 794)) or get a list of message board items for an object (GetAllMessageBoardEntries (see page 796)).
	Tags (see page 805)	The Tags class allows you to manage tags for content, users, and community groups. For example, you can add a tag to the system (AddTag (see page 806)), add a tag to a user (AddTagToUser (see page 811)), get a list of tags associated with a user (GetTagsForUser (see page 820)) or delete a tag by its ID (DeleteTagById (see page 812)).

CommunityGroup Class

The CommunityGroup class allows you to work with community groups in CMS400.NET. For example, you can add a community

group (AddCommunityGroup (see page 752)), get a list a list of community groups (GetAllCommunityGroups (see page 754)) or add a user to a community group (AddUserToCommunityGroup (see page 753)).

Class Hierarchy

C#

```
public class CommunityGroup : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class CommunityGroup
Inherits Ektron.Cms.API.Common
```

File

CommunityGroup.vb

CommunityGroup Methods

The methods of the CommunityGroup class are listed here.

Public Methods

	Name	Description
	AcceptGroupInvite (see page 752)	Accepts a group invitation on behalf of invited user.
	AddCommunityGroup (see page 752)	Create a new community group.
	AddUserToCommunityGroup (see page 752)	This is the overview for the AddUserToCommunityGroup method overload.
	ApprovePendingGroupUser (see page 753)	Approves the user's pending group request.
	CancelJoinRequestForCommunityGroup (see page 754)	Cancels a group join request sent by the user.
	DeleteCommunityGroupByID (see page 754)	Deletes an existing community group.
	DeletePendingGroupUser (see page 754)	Delete or reject a user's pending community group request.
	GetAllCommunityGroups (see page 754)	Returns an array of Community (see page 750) Groups based upon supplied criteria.
	GetAllCommunityGroupsInDirectory (see page 755)	Retrieves a list of community groups in a Directory (see page 196).
	GetAllUnassignedCommunityGroupUsers (see page 755)	Returns a list of users who are not currently in a community group.
	GetCommunityGroupBlog (see page 755)	Returns the blog folder ID associated with a community group.
	GetCommunityGroupBlogIds (see page 756)	Gets a list of Blog (see page 826) IDs that belong to a community group.
	GetCommunityGroupByID (see page 756)	Retrieves a CommunityGroupData object by ID.
	GetCommunityGroupDiscussionBoardIds (see page 756)	Gets a list of discussion board IDs that belong to a community group.
	GetCommunityGroups (see page 757)	Gets a list of community groups a user belongs to.
	GetCommunityGroupUserIDs (see page 757)	Gets a list of User (see page 950) IDs that belong to a community group.
	GetCommunityGroupUsers (see page 757)	This is the overview for the GetCommunityGroupUsers method overload.
	GetFolderIdForCommunityGroup (see page 758)	Gets the Folder (see page 683) ID used by the community group in the CMS content tree.
	GetGroupMemberStatus (see page 759)	This method allows you to find out status of a user for a particular group.
	GetInvitedCommunityGroups (see page 759)	Returns a list of community groups that a user has been invited to join.

	GetList (see page 759)	Returns a list of data for a given criteria.
	GetMostPopularCommunityGroups (see page 760)	This is the overview for the GetMostPopularCommunityGroups method overload.
	GetPendingCommunityGroups (see page 761)	Gets a list of community groups for which the user's membership is pending the group administrator's approval.
	GetPendingCommunityGroupUsers (see page 761)	Returns a list of users whose community group membership is pending approval.
	InviteUsers (see page 761)	Invites the users to the community group.
	IsCommunityGroupAdmin (see page 762)	Returns true if the user is an administrator of the group.
	IsUserInCommunityGroup (see page 762)	By providing the community group's ID and the user's ID, you can find out if a user belongs to a given community group. Returns true if the user is a member of the community group.
	RemoveUserFromCommunityGroup (see page 762)	Removes a user from a community group.
	UpdateCommunityGroup (see page 763)	Updates an existing community group.

CommunityGroup.AcceptGroupInvite Method

Accepts a group invitation on behalf of invited user.

C#

```
public AcceptGroupInvite(Long invitationId);
```

Visual Basic

```
Public Sub AcceptGroupInvite(ByVal invitationId As Long)
```

Parameters

Parameters	Description
ByVal invitationId As Long	Id of group invitation to accept.

CommunityGroup.AddCommunityGroup Method

Create a new community group.

C#

```
public Long AddCommunityGroup(CommunityGroupData groupData);
```

Visual Basic

```
Public Function AddCommunityGroup(ByVal groupData As CommunityGroupData) As Long
```

Parameters

Parameters	Description
ByVal groupData As CommunityGroupData	CommunityGroupData class containing information about the new community group.

Returns

ID of the new community group.

AddUserToCommunityGroup Method

This is the overview for the AddUserToCommunityGroup method overload.

Description

This method is overloaded. The first method allows you to add a user to a community group by providing the community group's ID and the user's ID. The second, allows you to add a user to a community group and override the group enrollment moderation by providing the community group's ID, the user's ID and setting the overridePending parameter to true.

Overload List

	Name	Description
	CommunityGroup.AddUserToCommunityGroup (Long, Long) (see page 753)	Adds a user to a community group.
	CommunityGroup.AddUserToCommunityGroup (Long, Long, Boolean) (see page 753)	Adds a user to a community group. You can override the group enrollment moderation by setting the overridePending parameter to true.

CommunityGroup.AddUserToCommunityGroup Method (Long, Long)

Adds a user to a community group.

C#

```
public AddUserToCommunityGroup(Long groupId, Long userId);
```

Visual Basic

```
Public Sub AddUserToCommunityGroup(ByVal groupId As Long, ByVal userId As Long)
```

Parameters

Parameters	Description
ByVal groupId As Long	ID of the community group to which the user will be added.
ByVal userId As Long	ID of the user to add.

Remarks

If the group does not have open enrollment, a group join request will be sent on behalf of the user.

CommunityGroup.AddUserToCommunityGroup Method (Long, Long, Boolean)

Adds a user to a community group. You can override the group enrollment moderation by setting the overridePending parameter to true.

C#

```
public AddUserToCommunityGroup(Long groupId, Long userId, Boolean overridePending);
```

Visual Basic

```
Public Sub AddUserToCommunityGroup(ByVal groupId As Long, ByVal userId As Long, ByVal
overridePending As Boolean)
```

Parameters

Parameters	Description
ByVal groupId As Long	ID of the community group to which the user will be added.
ByVal userId As Long	ID of the user to be added.
ByVal overridePending As Boolean	Set to true if you wish to bypass group enrollment moderation.

CommunityGroup.ApprovePendingGroupUser Method

Approves the user's pending group request.

C#

```
public ApprovePendingGroupUser(Long userId, Long groupId);
```

Visual Basic

```
Public Sub ApprovePendingGroupUser(ByVal userId As Long, ByVal groupId As Long)
```

Parameters

Parameters	Description
ByVal userId As Long	ID of user to approve.
ByVal groupId As Long	ID of the community group that the user is being approved to join.

CommunityGroup.CancelJoinRequestForCommunityGroup Method

Cancels a group join request sent by the user.

C#

```
public CancelJoinRequestForCommunityGroup(Long groupId, Long userId);
```

Visual Basic

```
Public Sub CancelJoinRequestForCommunityGroup(ByVal groupId As Long, ByVal userId As Long)
```

Parameters

Parameters	Description
ByVal groupId As Long	ID of the community group where the join request was sent.
ByVal userId As Long	ID of the user whom initially sent join request.

CommunityGroup.DeleteCommunityGroupByID Method

Deletes an existing community group.

C#

```
public DeleteCommunityGroupByID(Long groupId);
```

Visual Basic

```
Public Sub DeleteCommunityGroupByID(ByVal groupId As Long)
```

Parameters

Parameters	Description
ByVal groupId As Long	The ID of the community group to delete.

CommunityGroup.DeletePendingGroupUser Method

Delete or reject a user's pending community group request.

C#

```
public DeletePendingGroupUser(Long userId, Long groupId);
```

Visual Basic

```
Public Sub DeletePendingGroupUser(ByVal userId As Long, ByVal groupId As Long)
```

Parameters

Parameters	Description
ByVal userId As Long	ID of user whose request you are deleting .
ByVal groupId As Long	ID of community group from which the request is being deleted.

CommunityGroup.GetAllCommunityGroups Method

Returns an array of Community (see page 750) Groups based upon supplied criteria.

C#

```
public CommunityGroupData[] GetAllCommunityGroups(CommunityGroupRequest request);
```

Visual Basic

```
Public Function GetAllCommunityGroups(ByVal request As CommunityGroupRequest) As
CommunityGroupData()
```

Parameters

Parameters	Description
ByVal request As CommunityGroupRequest	request criteria defining the groups to retrieve.

Returns

An array of CommunityGroupData objects

CommunityGroup.GetAllCommunityGroupsInDirectory Method

Retrieves a list of community groups in a Directory (see page 196).

C#

```
public CommunityGroupData[] GetAllCommunityGroupsInDirectory(ref DirectoryRequest request);
```

Visual Basic

```
Public Function GetAllCommunityGroupsInDirectory(ByRef request As DirectoryRequest) As
CommunityGroupData()
```

Parameters

Parameters	Description
ByRef request As DirectoryRequest	Request criteria for community group retrieval.

Returns

An array of CommunityGroupData objects.

CommunityGroup.GetAllUnassignedCommunityGroupUsers Method

Returns a list of users who are not currently in a community group.

C#

```
public DirectoryUserData[] GetAllUnassignedCommunityGroupUsers(ref CommunityGroupObjectRequest
request);
```

Visual Basic

```
Public Function GetAllUnassignedCommunityGroupUsers(ByRef request As
CommunityGroupObjectRequest) As DirectoryUserData()
```

Parameters

Parameters	Description
ByRef request As CommunityGroupObjectRequest	Request criteria for user list.

Returns

An array of DirectoryUserData objects.

CommunityGroup.GetCommunityGroupBlog Method

Returns the blog folder ID associated with a community group.

C#

```
public Long GetCommunityGroupBlog(Long groupId);
```

Visual Basic

```
Public Function GetCommunityGroupBlog(ByVal groupId As Long) As Long
```

Parameters

Parameters	Description
ByVal groupId As Long	The community group ID.

Returns

A Blog (see page 826)'s folder ID.

CommunityGroup.GetCommunityGroupBlogIds Method

Gets a list of Blog (see page 826) IDs that belong to a community group.

C#

```
public Int64[] GetCommunityGroupBlogIds(Long communityGroupId);
```

Visual Basic

```
Public Function GetCommunityGroupBlogIds(ByVal communityGroupId As Long) As Int64()
```

Parameters

Parameters	Description
ByVal communityGroupId As Long	ID of community Group.

Returns

An array of blog IDs (Long).

CommunityGroup.GetCommunityGroupByID Method

Retrieves a CommunityGroupData object by ID.

C#

```
public CommunityGroupData GetCommunityGroupByID(Long groupId);
```

Visual Basic

```
Public Function GetCommunityGroupByID(ByVal groupId As Long) As CommunityGroupData
```

Parameters

Parameters	Description
ByVal groupId As Long	The ID of the community group to retrieve.

Returns

CommunityGroupData if the groupId is valid, else null.

CommunityGroup.GetCommunityGroupDiscussionBoardIds Method

Gets a list of discussion board IDs that belong to a community group.

C#

```
public Int64[] GetCommunityGroupDiscussionBoardIds(Long communityGroupId);
```

Visual Basic

```
Public Function GetCommunityGroupDiscussionBoardIds(ByVal communityGroupId As Long) As Int64()
```

Parameters

Parameters	Description
ByVal communityGroupId As Long	The ID of the community group.

Returns

An array of Discussion Board IDs (Long).

CommunityGroup.GetCommunityGroups Method

Gets a list of community groups a user belongs to.

C#

```
public CommunityGroupData[] GetCommunityGroups(Long userId, PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetCommunityGroups(ByVal userId As Long, ByVal pagingInfo As PagingInfo) As
CommunityGroupData()
```

Parameters

Parameters	Description
ByVal userId As Long	ID of user to retrieve community groups for.
ByVal pagingInfo As PagingInfo	Paging information for list retrieval.

Returns

An array of CommunityGroupData objects.

CommunityGroup.GetCommunityGroupUserIDs Method

Gets a list of User (see page 950) IDs that belong to a community group.

C#

```
public Long[] GetCommunityGroupUserIDs(Long communityGroupId);
```

Visual Basic

```
Public Function GetCommunityGroupUserIDs(ByVal communityGroupId As Long) As Long()
```

Parameters

Parameters	Description
ByVal communityGroupId As Long	The ID of the community group.

Returns

An array of Discussion Board IDs (Long).

GetCommunityGroupUsers Method

This is the overview for the GetCommunityGroupUsers method overload.

Description

This method is overloaded. The first method gets a list of users for a given community group. The second, gets a list of users for a given community group and gives you the ability to add paging to the list.

Overload List

	Name	Description
	CommunityGroup.GetCommunityGroupUsers (Long) (see page 758)	Retrieves a list of users in a community group.
	CommunityGroup.GetCommunityGroupUsers (Long, PagingInfo) (see page 758)	Returns a list of users in a community group.

CommunityGroup.GetCommunityGroupUsers Method (Long)

Retrieves a list of users in a community group.

C#

```
public DirectoryUserData[] GetCommunityGroupUsers(Long communityGroupId);
```

Visual Basic

```
Public Function GetCommunityGroupUsers(ByVal communityGroupId As Long) As DirectoryUserData()
```

Parameters

Parameters	Description
ByVal communityGroupId As Long	The ID of the community group.

Returns

An array of DirectoryUserData.

CommunityGroup.GetCommunityGroupUsers Method (Long, PagingInfo)

Returns a list of users in a community group.

C#

```
public DirectoryUserData[] GetCommunityGroupUsers(Long communityGroupId, PagingInfo
pagingInfo);
```

Visual Basic

```
Public Function GetCommunityGroupUsers(ByVal communityGroupId As Long, ByVal pagingInfo As
PagingInfo) As DirectoryUserData()
```

Parameters

Parameters	Description
ByVal communityGroupId As Long	The ID of the community group.
ByVal pagingInfo As PagingInfo	Paging information for the user list.

Returns

An array of DirectoryUserData objects.

CommunityGroup.GetFolderIdForCommunityGroup Method

Gets the Folder (see page 683) ID used by the community group in the CMS content tree.

C#

```
public Long GetFolderIdForCommunityGroup(Long CommunityGroupID);
```

Visual Basic

```
Public Function GetFolderIdForCommunityGroup(ByVal CommunityGroupID As Long) As Long
```

Parameters

Parameters	Description
ByVal CommunityGroupID As Long	ID of the community group.

Returns

A numeric value representing the folder ID.

CommunityGroup.GetGroupMemberStatus Method

This method allows you to find out status of a user for a particular group.

C#

```
public EkEnumeration.GroupMemberStatus GetGroupMemberStatus(Long communityGroupId, Long
userID);
```

Visual Basic

```
Public Function GetGroupMemberStatus(ByVal communityGroupId As Long, ByVal userID As Long) As
EkEnumeration.GroupMemberStatus
```

Parameters

Parameters	Description
ByVal communityGroupId As Long	ID of community group to get a user's status.
ByVal userID As Long	ID of the user.

Returns

GroupMemberStatus as an enumeration.

CommunityGroup.GetInvitedCommunityGroups Method

Returns a list of community groups that a user has been invited to join.

C#

```
public CommunityGroupData[] GetInvitedCommunityGroups(Long userId, PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetInvitedCommunityGroups(ByVal userId As Long, ByVal pagingInfo As
PagingInfo) As CommunityGroupData()
```

Parameters

Parameters	Description
ByVal userId As Long	ID of the user who was invited.
ByVal pagingInfo As PagingInfo	Object to supply paging parameters.

Returns

An array of CommunityGroupData.

CommunityGroup.GetList Method

Returns a list of data for a given criteria.

C#

```
public List(Of CommunityGroupData) GetList(Criteria(Of CommunityGroupProperty) criteria);
```

Visual Basic

```
Public Function GetList(ByVal criteria As Criteria(Of CommunityGroupProperty)) As List(Of
CommunityGroupData)
```

Parameters

Parameters	Description
ByVal criteria As Criteria(Of CommunityGroupProperty)	

GetMostPopularCommunityGroups Method

This is the overview for the GetMostPopularCommunityGroups method overload.

Description

This method is overloaded. The first method retrieves a list of the most popular community groups up to the number entered in the maxResults property. The second, retrieves all community groups ranked by popularity with the ability to have paging.

Note: The popularity of a group is determined by how many members belong to the group.

Overload List

	Name	Description
	CommunityGroup.GetMostPopularCommunityGroups (Integer) (see page 760)	Retrieves a list of the most popular community groups up to the number entered in the maxResults property. The popularity of a group is determined by how many members belong to the group.
	CommunityGroup.GetMostPopularCommunityGroups (PagingInfo) (see page 760)	Retrieves all community groups ranked by popularity with the ability to have paging. The popularity of a group is determined by how many members belong to the group.

CommunityGroup.GetMostPopularCommunityGroups Method (Integer)

Retrieves a list of the most popular community groups up to the number entered in the maxResults property. The popularity of a group is determined by how many members belong to the group.

C#

```
public CommunityGroupData[] GetMostPopularCommunityGroups(int maxResults);
```

Visual Basic

```
Public Function GetMostPopularCommunityGroups(ByVal maxResults As Integer) As
CommunityGroupData()
```

Parameters

Parameters	Description
ByVal maxResults As Integer	The number of groups to retrieve. For example, if 10 is supplied, the top 10 most popular groups will be returned.

Returns

An array of CommunityGroupData objects

CommunityGroup.GetMostPopularCommunityGroups Method (PagingInfo)

Retrieves all community groups ranked by popularity with the ability to have paging. The popularity of a group is determined by how many members belong to the group.

C#

```
public CommunityGroupData[] GetMostPopularCommunityGroups(PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetMostPopularCommunityGroups(ByVal pagingInfo As PagingInfo) As
CommunityGroupData()
```

Parameters

Parameters	Description
ByVal pagingInfo As PagingInfo	Paging information for paging through a large list of groups.

Returns

an array of CommunityGroupData objects

CommunityGroup.GetPendingCommunityGroups Method

Gets a list of community groups for which the user's membership is pending the group administrator's approval.

C#

```
public CommunityGroupData[] GetPendingCommunityGroups(Long userId, PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetPendingCommunityGroups(ByVal userId As Long, ByVal pagingInfo As
PagingInfo) As CommunityGroupData()
```

Parameters

Parameters	Description
ByVal userId As Long	ID of user to retrieve pending groups for.
ByVal pagingInfo As PagingInfo	Paging information for retrieval.

CommunityGroup.GetPendingCommunityGroupUsers Method

Returns a list of users whose community group membership is pending approval.

C#

```
public DirectoryUserData[] GetPendingCommunityGroupUsers(Long communityGroupID, PagingInfo
pagingInfo);
```

Visual Basic

```
Public Function GetPendingCommunityGroupUsers(ByVal communityGroupID As Long, ByVal pagingInfo
As PagingInfo) As DirectoryUserData()
```

Parameters

Parameters	Description
ByVal communityGroupID As Long	The ID of the community group.
ByVal pagingInfo As PagingInfo	Paging information for the user list.

Returns

An array of DirectoryUserData objects.

CommunityGroup.InviteUsers Method

Invites the users to the community group.

C#

```
public System.Collections.ObjectModel.Collection(Of Ektron.Cms.Community.Invitation)
InviteUsers(InvitationSendRequestData invitationData, Long groupId);
```

Visual Basic

```
Public Function InviteUsers(ByVal invitationData As InvitationSendRequestData, ByVal groupId
As Long) As System.Collections.ObjectModel.Collection(Of Ektron.Cms.Community.Invitation)
```

Parameters

Parameters	Description
ByVal invitationData As InvitationSendRequestData	InvitationSendRequestData containing information about the GroupInvitation to send.
ByVal groupId As Long	ID of the group to invite people to.

Returns

A collection of Invitations that were sent out.

CommunityGroup.IsCommunityGroupAdmin Method

Returns true if the user is an administrator of the group.

C#

```
public Boolean IsCommunityGroupAdmin(Long userId, Long groupId);
```

Visual Basic

```
Public Function IsCommunityGroupAdmin(ByVal userId As Long, ByVal groupId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal userId As Long	ID of the user to be verified.
ByVal groupId As Long	ID of the community group to check.

CommunityGroup.IsUserInCommunityGroup Method

By providing the community group's ID and the user's ID, you can find out if a user belongs to a given community group. Returns true if the user is a member of the community group.

C#

```
public Boolean IsUserInCommunityGroup(Long communityGroupId, Long userId);
```

Visual Basic

```
Public Function IsUserInCommunityGroup(ByVal communityGroupId As Long, ByVal userId As Long)
As Boolean
```

Parameters

Parameters	Description
ByVal communityGroupId As Long	ID of a community group to check membership.
ByVal userId As Long	User (see page 950) ID to be checked.

Returns

True if the user is a member of the community group.

CommunityGroup.RemoveUserFromCommunityGroup Method

Removes a user from a community group.

C#

```
public RemoveUserFromCommunityGroup(Long groupId, Long userId);
```

Visual Basic

```
Public Sub RemoveUserFromCommunityGroup(ByVal groupId As Long, ByVal userId As Long)
```

Parameters

Parameters	Description
ByVal groupId As Long	ID of the community group from which the user is being removed.
ByVal userId As Long	ID of the user to remove.

CommunityGroup.UpdateCommunityGroup Method

Updates an existing community group.

C#

```
public UpdateCommunityGroup(CommunityGroupData groupData);
```

Visual Basic

```
Public Sub UpdateCommunityGroup(ByVal groupData As CommunityGroupData)
```

Parameters

Parameters	Description
ByVal groupData As CommunityGroupData	CommunityGroupData class containing information to update for the group.

Favorites Class

The Favorites class allows you to work with a user's favorites. Favorites are content items that a user has designated as their favorite content. From this class, you can add content to a user's favorites (AddContentFavorite (see page 764)), remove content from a user's favorites (DeleteContentFavorite (see page 765)) or get a list of favorites for a given user (GetContentFavorites (see page 768)).

Class Hierarchy

C#

```
public class Favorites : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Favorites
Inherits Ektron.Cms.API.Common
```

File

Favorites.vb

Favorites Methods

The methods of the Favorites class are listed here.

Public Methods

	Name	Description
	AddContentFavorite (see page 764)	This is the overview for the AddContentFavorite method overload.
	AddFavoriteLink (see page 765)	Adds a URI to a specified user's favorites.
	DeleteContentFavorite (see page 765)	Deletes a piece of content form a user's favorites.
	DeleteFavoriteLink (see page 765)	Delete a URI from a user's favorites.
	GetAllFavoriteLink (see page 766)	This is the overview for the GetAllFavoriteLink method overload.
	GetContentFavorites (see page 767)	This is the overview for the GetContentFavorites method overload.

	GetFavoriteId (see page 769)	This method checks to see if a piece of the URI is already in a user's favorites. If it is, it returns the ID.
	GetFavoriteLink (see page 769)	Retrieve the favorite by providing the favorite's ID.
	GetFavoritesDirectory (see page 770)	Returns user's Favorites (see page 763) taxonomy directory structure.
	IsContentFavorite (see page 770)	Checks if to see if a piece of content is already in a user's favorites.
	MoveContentFavorite (see page 770)	Moves a content favorite to a new favorites directory.
	MoveFavoriteLink (see page 771)	Moves a favorite from one favorite directory to another favorite directory.
	UpdateFavoriteLink (see page 771)	Update an existing URI favorite by providing the favorite's ID, user's ID, URI name, URI description and actual URI.

AddContentFavorite Method

This is the overview for the AddContentFavorite method overload.

Description

This method is overloaded.

- AddContentFavorite (Long, Long) - adds a piece of content to user's favorites based on the content ID and user ID.
- AddContentFavorite (Long, Long, Long) - adds a piece of content to user's favorites and places it in a subdirectory based on the content ID, user ID and the ID of the subdirectory.

Overload List

	Name	Description
	Favorites.AddContentFavorite (Long, Long) (see page 764)	Adds a piece of content to user's favorites.
	Favorites.AddContentFavorite (Long, Long, Long) (see page 764)	Adds a piece of content to user's favorites and places it in a subdirectory.

Favorites.AddContentFavorite Method (Long, Long)

Adds a piece of content to user's favorites.

C#

```
public AddContentFavorite(Long contentId, Long userId);
```

Visual Basic

```
Public Sub AddContentFavorite(ByVal contentId As Long, ByVal userId As Long)
```

Parameters

Parameters	Description
ByVal contentId As Long	ID of content
ByVal userId As Long	ID of user

Favorites.AddContentFavorite Method (Long, Long, Long)

Adds a piece of content to user's favorites and places it in a subdirectory.

C#

```
public AddContentFavorite(Long contentId, Long userId, Long directoryId);
```

Visual Basic

```
Public Sub AddContentFavorite(ByVal contentId As Long, ByVal userId As Long, ByVal directoryId
As Long)
```

Parameters

Parameters	Description
ByVal contentId As Long	ID of the content item.
ByVal userId As Long	ID of user adding the favorite.
ByVal directoryId As Long	ID of the directory to add the content item to in the user's favorites.

Favorites.AddFavoriteLink Method

Adds a URI to a specified user's favorites.

C#

```
public Long AddFavoriteLink(Long userId, Long directoryId, int directoryLanguage, String
uriName, String Uri, String uriDescription);
```

Visual Basic

```
Public Function AddFavoriteLink(ByVal userId As Long, ByVal directoryId As Long, ByVal
directoryLanguage As Integer, ByVal uriName As String, ByVal Uri As String, ByVal
uriDescription As String) As Long
```

Parameters

Parameters	Description
ByVal userId As Long	The ID of the user.
ByVal directoryId As Long	A taxonomy ID.
ByVal directoryLanguage As Integer	A taxonomy language (for a user, this will be 0 - language neutral).
ByVal uriName As String	The name identification for the URI.
ByVal Uri As String	The actual URI.
ByVal uriDescription As String	A description of the URI.

Returns

The newly created favorite ID as a Long.

Favorites.DeleteContentFavorite Method

Deletes a piece of content form a user's favorites.

C#

```
public DeleteContentFavorite(Long contentId, Long userId);
```

Visual Basic

```
Public Sub DeleteContentFavorite(ByVal contentId As Long, ByVal userId As Long)
```

Parameters

Parameters	Description
ByVal contentId As Long	ID of the content item to be deleted from a user's favorites.
ByVal userId As Long	ID of user deleting the favorite.

Favorites.DeleteFavoriteLink Method

Delete a URI from a user's favorites.

C#

```
public DeleteFavoriteLink(Long favoriteId);
```

Visual Basic

```
Public Sub DeleteFavoriteLink(ByVal favoriteId As Long)
```

Parameters

Parameters	Description
ByVal favoriteId As Long	The ID of the favorite being deleted.

GetAllFavoriteLink Method

This is the overview for the GetAllFavoriteLink method overload.

Overload List

	Name	Description
	Favorites.GetAllFavoriteLink (Long) (see page 766)	Gets all the favorites for a given user.
	Favorites.GetAllFavoriteLink (Long, Long, Integer, Integer, Integer) (see page 766)	Gets all the favorites in user specified directory.
	Favorites.GetAllFavoriteLink (Long, Long, Integer, Integer, Integer, Integer) (see page 767)	Gets all the favorites in user specified directory.

Favorites.GetAllFavoriteLink Method (Long)

Gets all the favorites for a given user.

C#

```
public List(Of UriFavorite) GetAllFavoriteLink(Long userId);
```

Visual Basic

```
Public Function GetAllFavoriteLink(ByVal userId As Long) As List(Of UriFavorite)
```

Parameters

Parameters	Description
ByVal userId As Long	The user's ID.

Returns

A generic list of UriFavorite (see page 1646).

Favorites.GetAllFavoriteLink Method (Long, Long, Integer, Integer, Integer)

Gets all the favorites in user specified directory.

C#

```
public List(Of UriFavorite) GetAllFavoriteLink(Long userId, Long directoryId, int PageIndex,
int PageSize, ref int TotalPages);
```

Visual Basic

```
Public Function GetAllFavoriteLink(ByVal userId As Long, ByVal directoryId As Long, ByVal
PageIndex As Integer, ByVal PageSize As Integer, ByRef TotalPages As Integer) As List(Of
UriFavorite)
```

Parameters

Parameters	Description
ByVal userId As Long	The ID of the user.
ByVal directoryId As Long	The ID of the directory.

Returns

A generic list of UriFavorite (see page 1646).

Favorites.GetAllFavoriteLink Method (Long, Long, Integer, Integer, Integer, Integer)

Gets all the favorites in user specified directory.

C#

```
public List(Of UriFavorite) GetAllFavoriteLink(Long userId, Long directoryId, int PageIndex,
int PageSize, ref int TotalPages, ref int linkCount);
```

Visual Basic

```
Public Function GetAllFavoriteLink(ByVal userId As Long, ByVal directoryId As Long, ByVal
PageIndex As Integer, ByVal PageSize As Integer, ByRef TotalPages As Integer, ByRef linkCount
As Integer) As List(Of UriFavorite)
```

Parameters

Parameters	Description
ByVal userId As Long	The ID of the user.
ByVal directoryId As Long	The ID of the directory.
itemcount	total number of links in current and all its sub directories.

Returns

A generic list of UriFavorite (see page 1646).

GetContentFavorites Method

This is the overview for the GetContentFavorites method overload.

Description

This method is overloaded.

- GetContentFavorites (Long) - returns a list of the user's favorites based on a user ID.
- GetContentFavorites (Long, Long) - returns a list of the user's favorites in a directory based on the user ID and directory ID.
- GetContentFavorites (Long, Long, PagingInfo) - returns a list of the user's favorites in a given directory based on the user ID and directory ID. This method allows for paging.
- GetContentFavorites (Long, PagingInfo) - returns a list of the user's favorites based on a user ID. This method allows for paging.

Overload List

	Name	Description
	Favorites.GetContentFavorites (Long) (see page 768)	Returns a list of the user's favorites.
	Favorites.GetContentFavorites (Long, Long) (see page 768)	Returns a list of the user's favorites in a directory.

	Favorites.GetContentFavorites (Long, Long, PagingInfo) (see page 768)	Returns a list of the user's favorites in a given directory.
	Favorites.GetContentFavorites (Long, PagingInfo) (see page 769)	Returns a list of the user's favorites.

Favorites.GetContentFavorites Method (Long)

Returns a list of the user's favorites.

C#

```
public DirectoryContentData[] GetContentFavorites(Long userID);
```

Visual Basic

```
Public Function GetContentFavorites(ByVal userID As Long) As DirectoryContentData()
```

Parameters

Parameters	Description
ByVal userID As Long	ID of user to retrieve favorites for.

Returns

An array of DirectoryContentData objects.

Favorites.GetContentFavorites Method (Long, Long)

Returns a list of the user's favorites in a directory.

C#

```
public DirectoryContentData[] GetContentFavorites(Long userId, Long directoryId);
```

Visual Basic

```
Public Function GetContentFavorites(ByVal userId As Long, ByVal directoryId As Long) As
DirectoryContentData()
```

Parameters

Parameters	Description
ByVal userId As Long	ID of a user to retrieve favorites for.
ByVal directoryId As Long	ID of the directory to retrieve favorites from.

Returns

An array of DirectoryContentData objects

Favorites.GetContentFavorites Method (Long, Long, PagingInfo)

Returns a list of the user's favorites in a given directory.

C#

```
public DirectoryContentData[] GetContentFavorites(Long userId, Long directoryId, PagingInfo
pagingInfo);
```

Visual Basic

```
Public Function GetContentFavorites(ByVal userId As Long, ByVal directoryId As Long, ByVal
pagingInfo As PagingInfo) As DirectoryContentData()
```

Parameters

Parameters	Description
ByVal userId As Long	ID of the user to retrieve favorites for.
ByVal directoryId As Long	ID of the directory to retrieve favorites from.
ByVal pagingInfo As PagingInfo	Paging information for limiting the resultset.

Returns

An array of DirectoryContentData objects

Favorites.GetContentFavorites Method (Long, PagingInfo)

Returns a list of the user's favorites.

C#

```
public DirectoryContentData[] GetContentFavorites(Long userId, PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetContentFavorites(ByVal userId As Long, ByVal pagingInfo As PagingInfo) As
DirectoryContentData()
```

Parameters

Parameters	Description
ByVal userId As Long	ID of the user to retrieve favorites for.
ByVal pagingInfo As PagingInfo	Paging information for limiting resultset.

Returns

An array of DirectoryContentData objects.

Favorites.GetFavoriteId Method

This method checks to see if a piece of the URI is already in a user's favorites. If it is, it returns the ID.

C#

```
public Long GetFavoriteId(String favoriteName, String favoriteUri, Long userId);
```

Visual Basic

```
Public Function GetFavoriteId(ByVal favoriteName As String, ByVal favoriteUri As String, ByVal
userId As Long) As Long
```

Parameters

Parameters	Description
ByVal favoriteName As String	The favorite's name.
ByVal favoriteUri As String	The favorite's URI.
ByVal userId As Long	The ID of user to whose favorites are being checked.

Returns

If the favorite exists, it returns the ID; a non-negative value greater than 0.

Favorites.GetFavoriteLink Method

Retrieve the favorite by providing the favorite's ID.

C#

```
public UriFavorite GetFavoriteLink(Long favoriteId);
```

Visual Basic

```
Public Function GetFavoriteLink(ByVal favoriteId As Long) As UriFavorite
```

Parameters

Parameters	Description
ByVal favoriteId As Long	The ID of the favorite.

Returns

Returns information about the URI favorite in the format of UriFavorite (see page 1646).

Favorites.GetFavoritesDirectory Method

Returns user's Favorites (see page 763) taxonomy directory structure.

C#

```
public TaxonomyData GetFavoritesDirectory(Long userId, Long directoryId, PagingInfo
pagingInfo);
```

Visual Basic

```
Public Function GetFavoritesDirectory(ByVal userId As Long, ByVal directoryId As Long, ByVal
pagingInfo As PagingInfo) As TaxonomyData
```

Parameters

Parameters	Description
ByVal userId As Long	ID of the user who's favorites should be retrieved.
ByVal directoryId As Long	ID of the favorites directory to retrieve.
ByVal pagingInfo As PagingInfo	Paging information for limiting the resultset.

Favorites.IsContentFavorite Method

Checks if to see if a piece of content is already in a user's favorites.

C#

```
public Boolean IsContentFavorite(Long contentId, Long userId);
```

Visual Basic

```
Public Function IsContentFavorite(ByVal contentId As Long, ByVal userId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal contentId As Long	ID of content to be checked.
ByVal userId As Long	ID of user to check.

Returns

True if the content item is in user's favorites, else false.

Favorites.MoveContentFavorite Method

Moves a content favorite to a new favorites directory.

C#

```
public MoveContentFavorite(Long contentId, Long userId, Long directoryId);
```

Visual Basic

```
Public Sub MoveContentFavorite(ByVal contentId As Long, ByVal userId As Long, ByVal
directoryId As Long)
```

Parameters

Parameters	Description
ByVal contentId As Long	ID of the content item to move.
ByVal userId As Long	ID of the user.
ByVal directoryId As Long	ID of the directory to move content favorite it to.

Favorites.MoveFavoriteLink Method

Moves a favorite from one favorite directory to another favorite directory.

C#

```
public MoveFavoriteLink(Long favoriteId, Long sourceDir, Long destinationDir);
```

Visual Basic

```
Public Sub MoveFavoriteLink(ByVal favoriteId As Long, ByVal sourceDir As Long, ByVal
destinationDir As Long)
```

Parameters

Parameters	Description
ByVal favoriteId As Long	The ID of the favorite.
ByVal sourceDir As Long	The ID of the current directory.
ByVal destinationDir As Long	The ID of the new directory.

Favorites.UpdateFavoriteLink Method

Update an existing URI favorite by providing the favorite's ID, user's ID, URI name, URI description and actual URI.

C#

```
public UpdateFavoriteLink(Long favoriteId, Long userId, String uriName, String Uri, String
uriDescription);
```

Visual Basic

```
Public Sub UpdateFavoriteLink(ByVal favoriteId As Long, ByVal userId As Long, ByVal uriName As
String, ByVal Uri As String, ByVal uriDescription As String)
```

Parameters

Parameters	Description
ByVal favoriteId As Long	The ID of the favorite you are updating.
ByVal userId As Long	The ID of user whose favorite is being updated.
ByVal uriName As String	The name identifiction for the new URI.
ByVal Uri As String	The actual URI.
ByVal uriDescription As String	A description of the URI.

Flagging Class

The Flagging class allows you to work with Flag (see page 1231) Objects, Flag (see page 1231) Entries and content that has been flagged. For example, you can assign a flag to content (AssignFlagToContent (see page 772)) or get a list of flags for a content item (GetContentFlagData (see page 774)).

Class Hierarchy

C#

```
public class Flagging : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Flagging
Inherits Ektron.Cms.API.Common
```

File

Flagging.vb

Flagging Methods

The methods of the Flagging class are listed here.

Public Methods

	Name	Description
	AddFlagObject (see page 772)	Saves a new Flag (see page 1231) on an Object.
	AssignFlagToContent (see page 772)	Assigns a flag definition to a piece of content.
	GetAllFlagEntries (see page 773)	This is the overview for the GetAllFlagEntries method overload.
	GetContentFlagData (see page 774)	This is the overview for the GetContentFlagData method overload.
	GetDefaultFolderFlagDef (see page 775)	Retrieves all the flag definitions applied to a folder.
	GetFlagDataById (see page 775)	Retrieves a flag entry on a piece of content.
	GetFlaggedRank (see page 776)	Retrieves a flag ranking for an object.
	PurgeFlagEntries (see page 776)	Purges all flags of a particular content block over a defined date/time range.
	RemoveObjectFlag (see page 776)	Removes a flag entry from an object.
	UpdateFlagObject (see page 777)	Updates an existing Flag (see page 1231) on an Object.

Flagging.AddFlagObject Method

Saves a new Flag (see page 1231) on an Object.

C#

```
public Long AddFlagObject(ObjectFlagData objectFlag);
```

Visual Basic

```
Public Function AddFlagObject(ByVal objectFlag As ObjectFlagData) As Long
```

Parameters

Parameters	Description
ByVal objectFlag As ObjectFlagData	Object Flag (see page 1231) Data to save.

Returns

New Flag (see page 1231) Entry ID if successful, else 0.

Flagging.AssignFlagToContent Method

Assigns a flag definition to a piece of content.

C#

```
public AssignFlagToContent(Long contentId, Long flagId);
```

Visual Basic

```
Public Sub AssignFlagToContent(ByVal contentId As Long, ByVal flagId As Long)
```

Parameters

Parameters	Description
ByVal contentId As Long	ID of the content to be assigned the flag.
ByVal flagId As Long	ID of the flag definition.

GetAllFlagEntries Method

This is the overview for the GetAllFlagEntries method overload.

Description

This method is overloaded.

- GetAllFlagEntries (Integer, DateTime, DateTime, Integer) - gets a list of flag entries over a given time period based on the maximum number of results to retrieve, start and end dates and total flags.
- GetAllFlagEntries (Integer, Integer, DateTime, DateTime, Integer) - gets a list of flag entries over a given time period based on the content ID, maximum number of results to retrieve, start and end dates and total flags.

Overload List

	Name	Description
	Flagging.GetAllFlagEntries (Integer, DateTime, DateTime, Integer) (see page 773)	Gets a list of flag entries over a given time period.
	Flagging.GetAllFlagEntries (Long, Integer, DateTime, DateTime, Integer) (see page 774)	Gets a list of flag entries for a piece of content over a given time period.

Flagging.GetAllFlagEntries Method (Integer, DateTime, DateTime, Integer)

Gets a list of flag entries over a given time period.

C#

```
public ContentFlagData[] GetAllFlagEntries(int maxResults, DateTime startDate, DateTime
endDate, ref int totalFlags);
```

Visual Basic

```
Public Function GetAllFlagEntries(ByVal maxResults As Integer, ByVal startDate As DateTime,
ByVal endDate As DateTime, ByRef totalFlags As Integer) As ContentFlagData()
```

Parameters

Parameters	Description
ByVal maxResults As Integer	Max number of results to retrieve.
ByVal startDate As DateTime	Date to begin retrieving flags.
ByVal endDate As DateTime	Date to end retrieving flags.
ByRef totalFlags As Integer	Returns the total number of flags found.

Returns

Array of content flag data.

Flagging.GetAllFlagEntries Method (Long, Integer, DateTime, DateTime, Integer)

Gets a list of flag entries for a piece of content over a given time period.

C#

```
public ContentFlagData[] GetAllFlagEntries(Long contentId, int maxResults, DateTime startDate,
DateTime endDate, ref int totalFlags);
```

Visual Basic

```
Public Function GetAllFlagEntries(ByVal contentId As Long, ByVal maxResults As Integer, ByVal
startDate As DateTime, ByVal endDate As DateTime, ByRef totalFlags As Integer) As
ContentFlagData()
```

Parameters

Parameters	Description
ByVal contentId As Long	ID of content item for which to retrieve flags.
ByVal maxResults As Integer	Max number of results to retrieve.
ByVal startDate As DateTime	Date to begin retrieving flags.
ByVal endDate As DateTime	Date to end retrieving flags.
ByRef totalFlags As Integer	Returns the total number of flags found.

Returns

Array of content flag data.

GetContentFlagData Method

This is the overview for the GetContentFlagData method overload.

Description

This method is overloaded.

- GetContentFlagData (Long, Long) - returns flagData for a piece of content for a given user based the content's ID and user's ID.
- GetContentFlagData (Long, String) - returns flagData for a piece of content for a given user based the content's ID and non-cms user's visitor ID.

Overload List

	Name	Description
	Flagging.GetContentFlagData (Long, Long) (see page 774)	Returns flagData for a piece of content for a given user.
	Flagging.GetContentFlagData (Long, String) (see page 775)	Returns flagData for a piece of content for a given user.

Flagging.GetContentFlagData Method (Long, Long)

Returns flagData for a piece of content for a given user.

C#

```
public ContentFlagData GetContentFlagData(Long contentId, Long userId);
```

Visual Basic

```
Public Function GetContentFlagData(ByVal contentId As Long, ByVal userId As Long) As
```

```
ContentFlagData
```

Parameters

Parameters	Description
ByVal contentId As Long	ID of the content to retrieve flag.
ByVal userId As Long	ID of the CMS400.NET user who flagged content.

Returns

ContentFlagData object.

Flagging.GetContentFlagData Method (Long, String)

Returns flagData for a piece of content for a given user.

C#

```
public ContentFlagData GetContentFlagData(Long contentId, String visitorId);
```

Visual Basic

```
Public Function GetContentFlagData(ByVal contentId As Long, ByVal visitorId As String) As
ContentFlagData
```

Parameters

Parameters	Description
ByVal contentId As Long	ID of content to retrieve flag for.
ByVal visitorId As String	Visitor ID of non-cms user.

Returns

ContentFlagData object.

Flagging.GetDefaultFolderFlagDef Method

Retrieves all the flag definitions applied to a folder.

C#

```
public FolderFlagDefData GetDefaultFolderFlagDef(Long folderId);
```

Visual Basic

```
Public Function GetDefaultFolderFlagDef(ByVal folderId As Long) As FolderFlagDefData
```

Parameters

Parameters	Description
ByVal folderId As Long	ID of the folder to retrieve flag definitions.

Returns

array of FolderFlagDefData

Flagging.GetFlagDataById Method

Retrieves a flag entry on a piece of content.

C#

```
public ContentFlagData GetFlagDataById(Long flagId);
```

Visual Basic

```
Public Function GetFlagDataById(ByVal flagId As Long) As ContentFlagData
```

Parameters

Parameters	Description
ByVal flagId As Long	ID of the flag entry.

Returns

ContentFlagData

Flagging.GetFlaggedRank Method

Retrieves a flag ranking for an object.

C#

```
public FlaggedRankData[] GetFlaggedRank(Long ObjectId);
```

Visual Basic

```
Public Function GetFlaggedRank(ByVal ObjectId As Long) As FlaggedRankData()
```

Parameters

Parameters	Description
ByVal ObjectId As Long	ID of the object for which to retrieve a flag ranking.

Returns

FlaggedRankData

Flagging.PurgeFlagEntries Method

Purges all flags of a particular content block over a defined date/time range.

C#

```
public PurgeFlagEntries(Long objectId, DateTime startDate, DateTime endDate);
```

Visual Basic

```
Public Sub PurgeFlagEntries(ByVal objectId As Long, ByVal startDate As DateTime, ByVal endDate
As DateTime)
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the content that is having its flags purged.
ByVal startDate As DateTime	The beginning of the date range of content to be purged.
ByVal endDate As DateTime	The end of the date range of content to be purged.

Flagging.RemoveObjectFlag Method

Removes a flag entry from an object.

C#

```
public RemoveObjectFlag(Long flagId);
```

Visual Basic

```
Public Sub RemoveObjectFlag(ByVal flagId As Long)
```

Parameters

Parameters	Description
ByVal flagId As Long	ID of flag entry to remove.

Flagging.UpdateFlagObject Method

Updates an existing Flag (see page 1231) on an Object.

C#

```
public UpdateFlagObject(ObjectFlagData objectFlag);
```

Visual Basic

```
Public Sub UpdateFlagObject(ByVal objectFlag As ObjectFlagData)
```

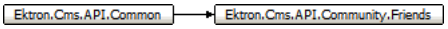
Parameters

Parameters	Description
ByVal objectFlag As ObjectFlagData	Object flag data to update based upon the FlagEntryId.

Friends Class

Friends, which are the same as Colleagues in CMS400.NET, allow users to associate themselves with other users in the community. This API (see page 669) class allows you to manage colleagues for users in the system. For example, you can add a colleague (AddFriend (see page 779)), remove a colleague (DeleteFriend (see page 780)) or get a list of current colleagues (GetFriends (see page 782)).

Class Hierarchy



C#

```
public class Friends : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Friends
Inherits Ektron.Cms.API.Common
```

File

Friends.vb

Friends Methods

The methods of the Friends class are listed here.

Public Methods

	Name	Description
	AcceptFriendInvite (see page 778)	Accepts a friend invitation on behalf of invited user.
	AcceptPendingFriend (see page 778)	Accepts a colleague request from one user to another.
	AddFriend (see page 778)	This is the overview for the AddFriend method overload.
	AddPendingFriend (see page 779)	Adds a pending colleague. Sends the other user a colleague invitation.
	AddSelectedFriend (see page 780)	Adds a colleague to a user's selected colleague list by providing the selected colleague's user ID and the ID of the user who wants to add the selected colleague.
	DeleteFriend (see page 780)	Delete's a colleague from a user's colleague list.
	DeletePendingFriendRequest (see page 780)	This is the overview for the DeletePendingFriendRequest method overload.
	DeleteSelectedFriend (see page 781)	Deletes a colleague from a user's selected colleague list by providing the selected colleague's user ID and the ID of the user who wants to delete the selected colleague.
	GetFriends (see page 781)	This is the overview for the GetFriends method overload.
	GetFriendsDirectory (see page 783)	Retrieves a user's colleague directory.

	GetFriendStatus (see page 784)	Get a user's colleague relation status to another user. For instance, user **A** may be a colleague to user **B**. User (see page 950) **C** may be a pending colleague to user **D**.
	GetInvitedUsers (see page 784)	Gets a list of users that a user has sent colleague invitations to.
	GetPendingFriends (see page 784)	Gets a list of users that have sent another user colleague invitations. This only returns users whose invitations have not yet been accepted and are still pending.
	GetSelectedFriends (see page 785)	Get a list of selected colleagues for a user by providing the user's ID.
	GetSentFriendInvitations (see page 785)	Gets a list of pending colleague invitations sent out by a user.
	InviteUsers (see page 785)	Send a friend invitation to a list of users and\or email addresses.
	IsFriend (see page 786)	Checks to see if a user is the colleague of another user by providing both colleague's ID information. In the friendUserID, enter the colleague's ID who you want to check. In the userID, enter the user you are checking against. Returns True when the users are colleagues. For example, if you want to find out whether Jim is a colleague of Tom, enter Jim's user ID in the friendUserId parameter and Tom's user ID in the userId parameter. If Jim and Tom are colleagues, this method returns true.
	IsSelectedFriend (see page 786)	Checks whether a user is a selected colleague of user.
	MoveFriend (see page 786)	Moves a colleague into another colleague directory.

Friends.AcceptFriendInvite Method

Accepts a friend invitation on behalf of invited user.

C#

```
public AcceptFriendInvite(Long invitationId);
```

Visual Basic

```
Public Sub AcceptFriendInvite(ByVal invitationId As Long)
```

Parameters

Parameters	Description
ByVal invitationId As Long	Id of friend invitation to accept.

Friends.AcceptPendingFriend Method

Accepts a colleague request from one user to another.

C#

```
public AcceptPendingFriend(Long fromUserId, Long toUserId);
```

Visual Basic

```
Public Sub AcceptPendingFriend(ByVal fromUserId As Long, ByVal toUserId As Long)
```

Parameters

Parameters	Description
ByVal fromUserId As Long	The ID of the user who sent the colleague request.
ByVal toUserId As Long	The ID of the user who received colleague request and is accepting it.

AddFriend Method

This is the overview for the AddFriend method overload.

Description

This function is overloaded. The first method allows you to add one user to another's colleague list. The second, allows you to add a user to a specified folder in another user's colleague list.

Overload List

	Name	Description
	Friends.AddFriend (Long, Long) (see page 779)	Adds a user to another user's colleague list.
	Friends.AddFriend (Long, Long, Long) (see page 779)	Adds a user to another user's friends list in a specific sub directory.

Friends.AddFriend Method (Long, Long)

Adds a user to another user's colleague list.

C#

```
public AddFriend(Long friendUserId, Long userId);
```

Visual Basic

```
Public Sub AddFriend(ByVal friendUserId As Long, ByVal userId As Long)
```

Parameters

Parameters	Description
ByVal friendUserId As Long	The ID of the user to be added to a user's colleague list.
ByVal userId As Long	The ID of the user adding the colleague.

Friends.AddFriend Method (Long, Long, Long)

Adds a user to another user's friends list in a specific sub directory.

C#

```
public AddFriend(Long friendUserId, Long userId, Long directoryId);
```

Visual Basic

```
Public Sub AddFriend(ByVal friendUserId As Long, ByVal userId As Long, ByVal directoryId As
Long)
```

Parameters

Parameters	Description
ByVal friendUserId As Long	The ID of the user to be added to the user's colleague list.
ByVal userId As Long	The ID of the user adding the colleague.
ByVal directoryId As Long	The ID of the directory where the new colleague will be added.

Friends.AddPendingFriend Method

Adds a pending colleague. Sends the other user a colleague invitation.

C#

```
public AddPendingFriend(Long friendUserId, Long userId);
```

Visual Basic

```
Public Sub AddPendingFriend(ByVal friendUserId As Long, ByVal userId As Long)
```

Parameters

Parameters	Description
ByVal friendUserId As Long	User (see page 950) receiving colleague request.
ByVal userId As Long	User (see page 950) sending colleague request.

Friends.AddSelectedFriend Method

Adds a colleague to a user's selected colleague list by providing the selected colleague's user ID and the ID of the user who wants to add the selected colleague.

C#

```
public AddSelectedFriend(Long selectedFriendUserId, Long userId);
```

Visual Basic

```
Public Sub AddSelectedFriend(ByVal selectedFriendUserId As Long, ByVal userId As Long)
```

Parameters

Parameters	Description
ByVal selectedFriendUserId As Long	User (see page 950) ID of a colleague to add to a user's selected colleague list.
ByVal userId As Long	ID of the user adding the selected colleague.

Friends.DeleteFriend Method

Delete's a colleague from a user's colleague list.

C#

```
public DeleteFriend(Long friendUserId, Long userId);
```

Visual Basic

```
Public Sub DeleteFriend(ByVal friendUserId As Long, ByVal userId As Long)
```

Parameters

Parameters	Description
ByVal friendUserId As Long	The user ID of the colleague to delete from the user's colleague list.
ByVal userId As Long	The ID of the user deleting the colleague.

DeletePendingFriendRequest Method

This is the overview for the DeletePendingFriendRequest method overload.

Description

This method is overloaded. The first method deletes a pending colleague request from a user's pending colleague list by providing fromUserId and toUserId information. The second, deletes a pending colleague from a user's pending colleague list by providing fromUserId and toEmailAddress information.

Overload List

	Name	Description
	Friends.DeletePendingFriendRequest (Long, Long) (see page 780)	Deletes a pending colleague request.
	Friends.DeletePendingFriendRequest (Long, String) (see page 781)	Deletes a pending colleague request.

Friends.DeletePendingFriendRequest Method (Long, Long)

Deletes a pending colleague request.

C#

```
public DeletePendingFriendRequest(Long fromUserId, Long toUserId);
```

Visual Basic

```
Public Sub DeletePendingFriendRequest(ByVal fromUserId As Long, ByVal toUserId As Long)
```

Parameters

Parameters	Description
ByVal fromUserId As Long	The ID of the user who sent the original colleague request.
ByVal toUserId As Long	The ID of the user who received the colleague request.

Friends.DeletePendingFriendRequest Method (Long, String)

Deletes a pending colleague request.

C#

```
public DeletePendingFriendRequest(Long fromUserId, String toEmailAddress);
```

Visual Basic

```
Public Sub DeletePendingFriendRequest(ByVal fromUserId As Long, ByVal toEmailAddress As String)
```

Parameters

Parameters	Description
ByVal fromUserId As Long	The ID of the user who sent original colleague request.
ByVal toEmailAddress As String	The Email address of the user who received the colleague request.

Friends.DeleteSelectedFriend Method

Deletes a colleague from a user's selected colleague list by providing the selected colleague's user ID and the ID of the user who wants to delete the selected colleague.

C#

```
public DeleteSelectedFriend(Long selectedFriendUserId, Long userId);
```

Visual Basic

```
Public Sub DeleteSelectedFriend(ByVal selectedFriendUserId As Long, ByVal userId As Long)
```

Parameters

Parameters	Description
ByVal selectedFriendUserId As Long	User (see page 950) ID of a colleague to delete from user's selected colleagues list.
ByVal userId As Long	ID of user deleting the colleague.

GetFriends Method

This is the overview for the GetFriends method overload.

Description

This function is overloaded.

- GetFriends (Long) - returns a list of colleagues based on a user's ID.
- GetFriends (Long, Long) - returns a list of colleagues in a specified directory based on a user's ID and the directory's ID.
- GetFriends (Long, PagingInfo) - returns a list of colleagues based on a user's ID. This method contains the ability to page results.
- GetFriends (Long, PagingInfo, Long) - returns a list of colleagues in a specified directory based on a user's ID and the directory's ID. This method contains the ability to page results.

Overload List

	Name	Description
	Friends.GetFriends (Long) (see page 782)	Returns a list of colleagues for a user.
	Friends.GetFriends (Long, Long) (see page 782)	Returns a list of colleagues in a specified directory for a given user.
	Friends.GetFriends (Long, PagingInfo) (see page 782)	Returns a list of colleagues for a user. This method contains the ability to page results.
	Friends.GetFriends (Long, PagingInfo, Long) (see page 783)	Returns a list of colleagues for a user in a specific directory. This method contains the ability to page the results.

Friends.GetFriends Method (Long)

Returns a list of colleagues for a user.

C#

```
public DirectoryUserData[] GetFriends(Long userId);
```

Visual Basic

```
Public Function GetFriends(ByVal userId As Long) As DirectoryUserData()
```

Parameters

Parameters	Description
ByVal userId As Long	The ID of the user whose colleague list is being returned.

Returns

An array of DirectoryUserData.

Friends.GetFriends Method (Long, Long)

Returns a list of colleagues in a specified directory for a given user.

C#

```
public DirectoryUserData[] GetFriends(Long userId, Long directoryID);
```

Visual Basic

```
Public Function GetFriends(ByVal userId As Long, ByVal directoryID As Long) As
DirectoryUserData()
```

Parameters

Parameters	Description
ByVal userId As Long	The ID of the user whose colleague list is being returned.
ByVal directoryID As Long	The ID of the directory from which to retrieve a colleague list.

Returns

An array of DirectoryUserData.

Friends.GetFriends Method (Long, PagingInfo)

Returns a list of colleagues for a user. This method contains the ability to page results.

C#

```
public DirectoryUserData[] GetFriends(Long userId, PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetFriends(ByVal userId As Long, ByVal pagingInfo As PagingInfo) As
DirectoryUserData()
```

Parameters

Parameters	Description
ByVal userId As Long	The ID of the user whose colleague list is being returned.
ByVal pagingInfo As PagingInfo	Supply paging information to control retrieval.

Returns

An array of DirectoryUserData.

Friends.GetFriends Method (Long, PagingInfo, Long)

Returns a list of colleagues for a user in a specific directory. This method contains the ability to page the results.

C#

```
public DirectoryUserData[] GetFriends(Long userId, PagingInfo pagingInfo, Long directoryID);
```

Visual Basic

```
Public Function GetFriends(ByVal userId As Long, ByVal pagingInfo As PagingInfo, ByVal
directoryID As Long) As DirectoryUserData()
```

Parameters

Parameters	Description
ByVal userId As Long	The ID of the user whose colleague list is being returned.
ByVal pagingInfo As PagingInfo	supply paging information to control retrieval.
ByVal directoryID As Long	The ID of the directory from which to retrieve a colleague list.

Returns

array of DirectoryUserData.

Friends.GetFriendsDirectory Method

Retrieves a user's colleague directory.

C#

```
public DirectoryAdvancedUserData GetFriendsDirectory(Long userId, Long directoryId, PagingInfo
pagingInfo);
```

Visual Basic

```
Public Function GetFriendsDirectory(ByVal userId As Long, ByVal directoryId As Long, ByVal
pagingInfo As PagingInfo) As DirectoryAdvancedUserData
```

Parameters

Parameters	Description
ByVal userId As Long	The ID of the user whose colleague directory is being retrieved.
ByVal directoryId As Long	ID of the colleague directory to retrieve.
ByVal pagingInfo As PagingInfo	Supply paging information to control retrieval.

Returns

DirectoryAdvancedUserData

Friends.GetFriendStatus Method

Get a user's colleague relation status to another user. For instance, user **A** may be a colleague to user **B**. User (see page 950) **C** may be a pending colleague to user **D**.

C#

```
public EkEnumeration.FriendStatus GetFriendStatus(Long friendUserId, Long userId);
```

Visual Basic

```
Public Function GetFriendStatus(ByVal friendUserId As Long, ByVal userId As Long) As
EkEnumeration.FriendStatus
```

Parameters

Parameters	Description
ByVal friendUserId As Long	The ID of the user whose status is being checked.
ByVal userId As Long	The ID of the user seeking status.

Returns

FriendStatus enumeration.

Friends.GetInvitedUsers Method

Gets a list of users that a user has sent colleague invitations to.

C#

```
public DirectoryUserData[] GetInvitedUsers(Long userId, PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetInvitedUsers(ByVal userId As Long, ByVal pagingInfo As PagingInfo) As
DirectoryUserData()
```

Parameters

Parameters	Description
ByVal userId As Long	ID of user who sent colleague invitations.
ByVal pagingInfo As PagingInfo	Supply paging information to control retrieval.

Returns

An array of DirectoryUserData.

Friends.GetPendingFriends Method

Gets a list of users that have sent another user colleague invitations. This only returns users whose invitations have not yet been accepted and are still pending.

C#

```
public DirectoryUserData[] GetPendingFriends(Long userId, PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetPendingFriends(ByVal userId As Long, ByVal pagingInfo As PagingInfo) As
DirectoryUserData()
```

Parameters

Parameters	Description
ByVal userId As Long	ID of user who has received invitations.
ByVal pagingInfo As PagingInfo	Supply paging information to control retrieval.

Returns

array of DirectoryUserData

Friends.GetSelectedFriends Method

Get a list of selected colleagues for a user by providing the user's ID.

C#

```
public DirectoryUserData[] GetSelectedFriends(Long userId);
```

Visual Basic

```
Public Function GetSelectedFriends(ByVal userId As Long) As DirectoryUserData()
```

Parameters

Parameters	Description
ByVal userId As Long	ID of the user to retrieve selected colleagues.

Returns

An array of DirectoryUserData.

Friends.GetSentFriendInvitations Method

Gets a list of pending colleague invitations sent out by a user.

C#

```
public FriendInvitation[] GetSentFriendInvitations(Long userId, PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetSentFriendInvitations(ByVal userId As Long, ByVal pagingInfo As PagingInfo)
As FriendInvitation()
```

Parameters

Parameters	Description
ByVal userId As Long	ID of the user who sent invitations.
ByVal pagingInfo As PagingInfo	Supply paging information to control retrieval.

Returns

AN array of FriendInvitations.

Friends.InviteUsers Method

Send a friend invitation to a list of users and\or email addresses.

C#

```
public ObjectModel.Collection(Of Ektron.Cms.Community.Invitation)
InviteUsers(InvitationSendRequestData invitationData);
```

Visual Basic

```
Public Function InviteUsers(ByVal invitationData As InvitationSendRequestData) As
ObjectModel.Collection(Of Ektron.Cms.Community.Invitation)
```

Parameters

Parameters	Description
ByVal invitationData As InvitationSendRequestData	Invitation request data detailing invitation information.

Returns

A collection of Invitation objects that have been sent out.

Friends.IsFriend Method

Checks to see if a user is the colleague of another user by providing both colleague's ID information. In the friendUserID, enter the colleague's ID who you want to check. In the userID, enter the user you are checking against. Returns True when the users are colleagues. For example, if you want to find out whether Jim is a colleague of Tom, enter Jim's user ID in the friendUserId parameter and Tom's user ID in the userId parameter. If Jim and Tom are colleagues, this method returns true.

C#

```
public Boolean IsFriend(Long friendUserId, Long userId);
```

Visual Basic

```
Public Function IsFriend(ByVal friendUserId As Long, ByVal userId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal friendUserId As Long	first userId
ByVal userId As Long	second userId

Returns

Returns true if FriendUserId and UserID are colleagues, else false.

Friends.IsSelectedFriend Method

Checks whether a user is a selected colleague of user.

C#

```
public Boolean IsSelectedFriend(Long friendUserId, Long userId);
```

Visual Basic

```
Public Function IsSelectedFriend(ByVal friendUserId As Long, ByVal userId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal friendUserId As Long	User (see page 950) ID to check in selected colleagues list.
ByVal userId As Long	ID of the user who is checking for a selected colleague.

Returns

Returns true if the colleague's user ID is in the selected colleagues list of for a user ID, else false.

Friends.MoveFriend Method

Moves a colleague into another colleague directory.

C#

```
public MoveFriend(Long friendUserId, Long userId, Long directoryId);
```

Visual Basic

```
Public Sub MoveFriend(ByVal friendUserId As Long, ByVal userId As Long, ByVal directoryId As
Long)
```

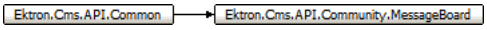
Parameters

Parameters	Description
ByVal friendUserId As Long	The user ID of the colleague to be moved.
ByVal userId As Long	The ID of the user moving the colleague.
ByVal directoryId As Long	The ID of the directory where the colleague will be moved.

MessageBoard Class

The MessageBoard class allows you to work with message board entries associated with a user, community group or content item. For example, you can add a message to a message board (AddMessageBoardEntry (see page 788)), delete a message (DeleteMessageBoardEntry (see page 794)) or get a list of message board items for an object (GetAllMessageBoardEntries (see page 796)).

Class Hierarchy



C#

```
public class MessageBoard : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class MessageBoard
Inherits Ektron.Cms.API.Common
```

File

MessageBoard.vb

MessageBoard Methods

The methods of the MessageBoard class are listed here.

Public Methods

	Name	Description
	AddMessageBoardEntry (see page 788)	Adds a Message Board Entry
	ApproveMessage (see page 788)	This is the overview for the ApproveMessage method overload.
	DeleteMessageBoardEntries (see page 790)	This is the overview for the DeleteMessageBoardEntries method overload.
	DeleteMessageBoardEntriesByUser (see page 792)	This is the overview for the DeleteMessageBoardEntriesByUser method overload.
	DeleteMessageBoardEntry (see page 793)	This is the overview for the DeleteMessageBoardEntry method overload.
	DisableModeration (see page 795)	Switch off Moderation for the Object.
	EnableModeration (see page 795)	Switch on Moderation for the Object.
	GetAllMessageBoardEntries (see page 796)	This is the overview for the GetAllMessageBoardEntries method overload.
	GetAllMessageBoardEntriesByUser (see page 798)	This is the overview for the GetAllMessageBoardEntriesByUser method overload.
	GetMessageBoardEntry (see page 800)	This is the overview for the GetMessageBoardEntry method overload.
	IsModerated (see page 802)	Check if the object is moderated or not.
	IsSpam (see page 802)	This function evaluates if the new message is spam or not based on the spam filter selected.
	IsUserMessageBoardAdmin (see page 802)	This function determines if the user has administrative privileges for the message board.
	UpdateMessageText (see page 803)	This is the overview for the UpdateMessageText method overload.

MessageBoard.AddMessageBoardEntry Method

Adds a Message Board Entry

C#

```
public MessageBoardData AddMessageBoardEntry(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, String messageText, Boolean isModerated, Long
callerUserId);
```

Visual Basic

```
Public Function AddMessageBoardEntry(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal messageText As String, ByVal isModerated As
Boolean, ByVal callerUserId As Long) As MessageBoardData
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object that is associated with the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.
ByVal messageText As String	the text of the message being added to the board.
ByVal isModerated As Boolean	The moderation switch.
ByVal callerUserId As Long	The user ID of the member requesting data.

ApproveMessage Method

This is the overview for the ApproveMessage method overload.

Description

This method is overloaded.

- ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType) - change the flag of the message from pending to approved by providing the message ID, object ID and type.
- ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType, Long) - change the flag of the message from pending to approved by providing the message ID, object ID and type, and the ID of the user who changing the flag.
- ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) - change the flag of the message from pending to approved by providing the message ID, object ID and type, the ID of the user who changing the flag and whether the message board is moderated.

Overload List

	Name	Description
	MessageBoard.ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType) (see page 789)	This function changes the flag of the message from pending to approve.
	MessageBoard.ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType, Long) (see page 789)	This function changes the flag of the message from pending to approve.
	MessageBoard.ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) (see page 789)	This function changes the flag of the message from pending to approve.

MessageBoard.ApproveMessage Method (Long, Long, EkEnumeration.MessageBoardObjectType)

This function changes the flag of the message from pending to approve.

C#

```
public MessageBoardData ApproveMessage(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType);
```

Visual Basic

```
Public Function ApproveMessage(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType) As MessageBoardData
```

Parameters

Parameters	Description
ByVal messageId As Long	The unique ID of the message.
ByVal objectId As Long	The ID of the object.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of the object.

Returns

Returns the updated row.

MessageBoard.ApproveMessage Method (Long, Long, EkEnumeration.MessageBoardObjectType, Long)

This function changes the flag of the message from pending to approve.

C#

```
public MessageBoardData ApproveMessage(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long callerUserId);
```

Visual Basic

```
Public Function ApproveMessage(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long) As
MessageBoardData
```

Parameters

Parameters	Description
ByVal messageId As Long	The unique ID of the message.
ByVal objectId As Long	The ID of the object associated with the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of the object associated with the message board.
ByVal callerUserId As Long	The user ID of the member doing the update.

Returns

Returns the updated row.

MessageBoard.ApproveMessage Method (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean)

This function changes the flag of the message from pending to approve.

C#

```
public MessageBoardData ApproveMessage(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long callerUserId, Boolean isModerated);
```

Visual Basic

```
Public Function ApproveMessage(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long, ByVal
```

```
isModerated As Boolean) As MessageBoardData
```

Parameters

Parameters	Description
ByVal messageId As Long	The unique ID of the message.
ByVal objectId As Long	The ID of the object associated with the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of the object associated with the message board.
ByVal callerUserId As Long	The user ID of the member doing the update.
ByVal isModerated As Boolean	The moderation switch.

Returns

Returns the updated row.

DeleteMessageBoardEntries Method

This is the overview for the DeleteMessageBoardEntries method overload.

Description

This method is overloaded.

- DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType) - deletes all the message board records for a particular object based on the object ID and type.
- DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, Long) - deletes all the message board records for a particular object based on the object ID and type, and the user ID of the member deleting the messages.
- DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) - deletes all the message board records for a particular object based on the object ID and type, the user ID of the member deleting the messages, and whether the message board is moderated.

Overload List

	Name	Description
	MessageBoard.DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType) (see page 790)	Deletes all the message board records for a particular object.
	MessageBoard.DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, Long) (see page 791)	Deletes all the message board records for a particular object.
	MessageBoard.DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) (see page 791)	Deletes all the message board records for a particular object.

MessageBoard.DeleteMessageBoardEntries Method (Long, EkEnumeration.MessageBoardObjectType)

Deletes all the message board records for a particular object.

C#

```
public Boolean DeleteMessageBoardEntries(Long objectId, EkEnumeration.MessageBoardObjectType
objectType);
```

Visual Basic

```
Public Function DeleteMessageBoardEntries(ByVal objectId As Long, ByVal objectType As
```

```
EkEnumeration.MessageBoardObjectType) As Boolean
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.

Returns

Returns true when all the messages are deleted successfully.

MessageBoard.DeleteMessageBoardEntries Method (Long, EkEnumeration.MessageBoardObjectType, Long)

Deletes all the message board records for a particular object.

C#

```
public Boolean DeleteMessageBoardEntries(Long objectId, EkEnumeration.MessageBoardObjectType
objectType, Long callerUserId);
```

Visual Basic

```
Public Function DeleteMessageBoardEntries(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.
ByVal callerUserId As Long	The user ID of the member deleting the messages.

Returns

Returns true when all the messages are deleted successfully.

MessageBoard.DeleteMessageBoardEntries Method (Long, EkEnumeration.MessageBoardObjectType, Long, Boolean)

Deletes all the message board records for a particular object.

C#

```
public Boolean DeleteMessageBoardEntries(Long objectId, EkEnumeration.MessageBoardObjectType
objectType, Long callerUserId, Boolean isModerated);
```

Visual Basic

```
Public Function DeleteMessageBoardEntries(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long, ByVal isModerated As
Boolean) As Boolean
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.
ByVal callerUserId As Long	The user ID of the member deleting the messages.
ByVal isModerated As Boolean	The moderation switch.

Returns

Returns true when all the messages are deleted successfully

DeleteMessageBoardEntriesByUser Method

This is the overview for the DeleteMessageBoardEntriesByUser method overload.

Description

This method is overloaded.

- DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long) - deletes all of a user's messages on a given message board based on the object ID and type, and the user's ID.
- DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, Long) - deletes all of a user's messages on a given message board based on the object ID and type, the ID of the user whose messages are being removed and the ID of the user removing the messages.
- DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, Long, Boolean) - deletes all of a user's messages on a given message board based on the object ID and type, the ID of the user whose messages are being removed, the ID of the user removing the messages and whether the message board is moderated.

Overload List

	Name	Description
	MessageBoard.DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long) (see page 792)	Deletes all the message board records for a particular User (see page 950).
	MessageBoard.DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, Long) (see page 792)	Deletes all the message board records for a particular User (see page 950).
	MessageBoard.DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, Long, Boolean) (see page 793)	Deletes all the message board records for a particular User (see page 950).

MessageBoard.DeleteMessageBoardEntriesByUser Method (Long, EkEnumeration.MessageBoardObjectType, Long)

Deletes all the message board records for a particular User (see page 950).

C#

```
public Boolean DeleteMessageBoardEntriesByUser(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long userId);
```

Visual Basic

```
Public Function DeleteMessageBoardEntriesByUser(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.
ByVal userId As Long	The ID of the user whose messages are going to be deleted.

MessageBoard.DeleteMessageBoardEntriesByUser Method (Long, EkEnumeration.MessageBoardObjectType, Long, Long)

Deletes all the message board records for a particular User (see page 950).

C#

```
public Boolean DeleteMessageBoardEntriesByUser(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long userId, Long callerUserId);
```

Visual Basic

```
Public Function DeleteMessageBoardEntriesByUser(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long, ByVal callerUserId As Long) As
Boolean
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.
ByVal userId As Long	The ID of the user whose messages are going to be deleted.
ByVal callerUserId As Long	The user ID of the member deleting the messages.

MessageBoard.DeleteMessageBoardEntriesByUser Method (Long, EkEnumeration.MessageBoardObjectType, Long, Long, Boolean)

Deletes all the message board records for a particular User (see page 950).

C#

```
public Boolean DeleteMessageBoardEntriesByUser(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long userId, Long callerUserId, Boolean
isModerated);
```

Visual Basic

```
Public Function DeleteMessageBoardEntriesByUser(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long, ByVal callerUserId As Long, ByVal
isModerated As Boolean) As Boolean
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.
ByVal userId As Long	The ID of the user whose messages are going to be deleted.
ByVal callerUserId As Long	The user ID of the member deleting the messages.
ByVal isModerated As Boolean	The moderation switch.

DeleteMessageBoardEntry Method

This is the overview for the DeleteMessageBoardEntry method overload.

Description

This method is overloaded.

- DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType) - deletes single message board record based on the message ID, object ID and object type.
- DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long) - deletes single message board record based on the message ID, object ID, object type and user ID requesting the data.
- DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) - deletes single message board record based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

Overload List

	Name	Description
	MessageBoard.DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType) (see page 794)	Deletes single message board record based on the message ID, object ID and object type.
	MessageBoard.DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long) (see page 794)	Deletes single message board record based on the message ID, object ID, object type and user ID requesting the data.
	MessageBoard.DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) (see page 795)	Deletes single message board record based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

MessageBoard.DeleteMessageBoardEntry Method (Long, Long, EkEnumeration.MessageBoardObjectType)

Deletes single message board record based on the message ID, object ID and object type.

C#

```
public Boolean DeleteMessageBoardEntry(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType);
```

Visual Basic

```
Public Function DeleteMessageBoardEntry(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType) As Boolean
```

Parameters

Parameters	Description
ByVal messageId As Long	The unique ID of the message.
ByVal objectId As Long	The ID of the object that is associated with the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.

Returns

Returns true when the message is deleted successfully.

MessageBoard.DeleteMessageBoardEntry Method (Long, Long, EkEnumeration.MessageBoardObjectType, Long)

Deletes single message board record based on the message ID, object ID, object type and user ID requesting the data.

C#

```
public Boolean DeleteMessageBoardEntry(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long callerUserId);
```

Visual Basic

```
Public Function DeleteMessageBoardEntry(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal messageId As Long	The unique ID of the message.
ByVal objectId As Long	The ID of the object that is associated with the message board.

ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.
ByVal callerUserId As Long	The user ID of the member deleting the message.

Returns

Returns true when the message is deleted successfully.

MessageBoard.DeleteMessageBoardEntry Method (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean)

Deletes single message board record based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

C#

```
public Boolean DeleteMessageBoardEntry(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long callerUserId, Boolean isModerated);
```

Visual Basic

```
Public Function DeleteMessageBoardEntry(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long, ByVal
isModerated As Boolean) As Boolean
```

Parameters

Parameters	Description
ByVal messageId As Long	The unique ID of the message.
ByVal objectId As Long	The ID of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.
ByVal callerUserId As Long	The user ID of the member deleting the message.
ByVal isModerated As Boolean	The moderation switch.

Returns

Returns true when the message is deleted successfully.

MessageBoard.DisableModeration Method

Switch off Moderation for the Object.

C#

```
public DisableModeration(Long objectId, EkEnumeration.MessageBoardObjectType objectType, Long
callerUserId);
```

Visual Basic

```
Public Sub DisableModeration(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long)
```

Parameters

Parameters	Description
ByVal objectId As Long	The id of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.

MessageBoard.EnableModeration Method

Switch on Moderation for the Object.

C#

```
public EnableModeration(Long objectId, EkEnumeration.MessageBoardObjectType objectType, Long
```

```
callerUserId);
```

Visual Basic

```
Public Sub EnableModeration(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long)
```

Parameters

Parameters	Description
ByVal objectId As Long	The id of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.

GetAllMessageBoardEntries Method

This is the overview for the GetAllMessageBoardEntries method overload.

Description

This method is overloaded.

- GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo) - gets all messages for a particular object based on the object ID, object type and paging information.
- GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long) - gets all messages for a particular object based on the object ID, object type, ID of the user requesting the list and paging information.
- GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long, Boolean) - gets all messages for a particular object based on the object ID, object type, ID of the user requesting the list, paging information and the moderation switch.

Overload List

	Name	Description
	MessageBoard.GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo) (see page 796)	Gets all messages for a particular object.
	MessageBoard.GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long) (see page 797)	Gets all messages for a particular object.
	MessageBoard.GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long, Boolean) (see page 797)	Gets all messages for a particular object.

MessageBoard.GetAllMessageBoardEntries Method (Long, EkEnumeration.MessageBoardObjectType, PagingInfo)

Gets all messages for a particular object.

C#

```
public Collection(Of MessageBoardData) GetAllMessageBoardEntries(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, PagingInfo paging);
```

Visual Basic

```
Public Function GetAllMessageBoardEntries(ByVal objectId As Long, ByVal objectType As
```

```
EkEnumeration.MessageBoardObjectType, ByVal paging As PagingInfo) As Collection(Of
MessageBoardData)
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of the object.
ByVal paging As PagingInfo	The paging object. Setting "RecordsPerPage" to zero retrieves all rows.

Returns

Returns a collection of message board data objects for each record retrieved from the database.

MessageBoard.GetAllMessageBoardEntries Method (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long)

Gets all messages for a particular object.

C#

```
public Collection(Of MessageBoardData) GetAllMessageBoardEntries(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, PagingInfo paging, Long callerUserId);
```

Visual Basic

```
Public Function GetAllMessageBoardEntries(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal paging As PagingInfo, ByVal callerUserId As Long)
As Collection(Of MessageBoardData)
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of the object.
ByVal paging As PagingInfo	The paging object. Setting "RecordsPerPage" to zero retrieves all rows.
ByVal callerUserId As Long	The user ID of the member requesting data.

Returns

Returns a collection of message board data objects for each record retrieved from the database.

MessageBoard.GetAllMessageBoardEntries Method (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long, Boolean)

Gets all messages for a particular object.

C#

```
public Collection(Of MessageBoardData) GetAllMessageBoardEntries(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, PagingInfo paging, Long callerUserId, Boolean
isModerated);
```

Visual Basic

```
Public Function GetAllMessageBoardEntries(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal paging As PagingInfo, ByVal callerUserId As Long,
ByVal isModerated As Boolean) As Collection(Of MessageBoardData)
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of the object.

ByVal paging As PagingInfo	The paging object. Setting "RecordsPerPage" to zero retrieves all rows.
ByVal callerUserId As Long	The user ID of the member requesting data.
ByVal isModerated As Boolean	The moderation switch.

Returns

Returns a collection of message board data objects for each record retrieved from the database.

GetAllMessageBoardEntriesByUser Method

This is the overview for the GetAllMessageBoardEntriesByUser method overload.

Description

This method is overloaded.

- GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Integer, PagingInfo) - gets all messages entered by a user on a given message board based on the object ID and type, the user's ID and paging information.
- GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long) - gets all messages entered by a user on a given message board based on the object ID and type, the ID of the user whose messages are being removed, the ID of the user removing the messages and paging information.
- GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long, Boolean) - gets all messages entered by a user on a given message board based on the object ID and type, the ID of the user whose messages are being removed, the ID of the user removing the messages and whether the message board is moderated.

Overload List

	Name	Description
	MessageBoard.GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo) (see page 798)	Gets all messages entered by a user.
	MessageBoard.GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long) (see page 799)	Gets all messages entered by a user.
	MessageBoard.GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long, Boolean) (see page 799)	Gets all messages entered by a user.

MessageBoard.GetAllMessageBoardEntriesByUser Method (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo)

Gets all messages entered by a user.

C#

```
public Collection(Of MessageBoardData) GetAllMessageBoardEntriesByUser(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long userId, PagingInfo paging);
```

Visual Basic

```
Public Function GetAllMessageBoardEntriesByUser(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long, ByVal paging As PagingInfo) As
Collection(Of MessageBoardData)
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object.

ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of the object.
ByVal userId As Long	The user ID for which to retrieve data.
ByVal paging As PagingInfo	The paging object. Setting "RecordsPerPage" to zero retrieves all rows.

Returns

Returns a collection of message board data objects for each record retrieved from the database.

MessageBoard.GetAllMessageBoardEntriesByUser Method (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long)

Gets all messages entered by a user.

C#

```
public Collection(Of MessageBoardData) GetAllMessageBoardEntriesByUser(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long userId, PagingInfo paging, Long
callerUserId);
```

Visual Basic

```
Public Function GetAllMessageBoardEntriesByUser(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long, ByVal paging As PagingInfo, ByVal
callerUserId As Long) As Collection(Of MessageBoardData)
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of the object.
ByVal userId As Long	The user ID for which to retrieve data.
ByVal paging As PagingInfo	The paging object. Setting "RecordsPerPage" to zero retrieves all rows.
ByVal callerUserId As Long	The user ID of the member requesting data.

Returns

Returns a collection of message board data objects for each record retrieved from the database.

MessageBoard.GetAllMessageBoardEntriesByUser Method (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long, Boolean)

Gets all messages entered by a user.

C#

```
public Collection(Of MessageBoardData) GetAllMessageBoardEntriesByUser(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long userId, PagingInfo paging, Long
callerUserId, Boolean isModerated);
```

Visual Basic

```
Public Function GetAllMessageBoardEntriesByUser(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long, ByVal paging As PagingInfo, ByVal
callerUserId As Long, ByVal isModerated As Boolean) As Collection(Of MessageBoardData)
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of the object.
ByVal userId As Long	The user ID for which to retrieve data.
ByVal paging As PagingInfo	The paging object. Setting "RecordsPerPage" to zero retrieves all rows.
ByVal callerUserId As Long	The user ID of the member requesting data.

ByVal isModerated As Boolean	The moderation switch.

Returns

Returns a collection of message board data objects for each record retrieved from the database.

GetMessageBoardEntry Method

This is the overview for the GetMessageBoardEntry method overload.

Description

This method is overload.

GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType) - gets a specific message from a message board based on the message ID, object ID and object type.

GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long) - gets a specific message from a message board based on the message ID, object ID, object type and user ID requesting the data.

GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) - gets a specific message from a message board based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

Overload List

	Name	Description
	MessageBoard.GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType) (see page 800)	Gets a specific message from a message board based on the message ID, object ID and object type.
	MessageBoard.GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long) (see page 801)	Gets a specific message from a message board based on the message ID, object ID, object type and user ID requesting the data.
	MessageBoard.GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) (see page 801)	Gets a specific message from a message board based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

MessageBoard.GetMessageBoardEntry Method (Long, Long, EkEnumeration.MessageBoardObjectType)

Gets a specific message from a message board based on the message ID, object ID and object type.

C#

```
public MessageBoardData GetMessageBoardEntry(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType);
```

Visual Basic

```
Public Function GetMessageBoardEntry(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType) As MessageBoardData
```

Parameters

Parameters	Description
ByVal messageId As Long	The unique ID of the message.
ByVal objectId As Long	The ID of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.

Returns

The message board data object with the data from the database.

MessageBoard.GetMessageBoardEntry Method (Long, Long, EkEnumeration.MessageBoardObjectType, Long)

Gets a specific message from a message board based on the message ID, object ID, object type and user ID requesting the data.

C#

```
public MessageBoardData GetMessageBoardEntry(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long callerUserId);
```

Visual Basic

```
Public Function GetMessageBoardEntry(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long) As
MessageBoardData
```

Parameters

Parameters	Description
ByVal messageId As Long	The unique ID of the message.
ByVal objectId As Long	The ID of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.
ByVal callerUserId As Long	The user ID of the member requesting data.

Returns

The message board data object with the data from the database.

MessageBoard.GetMessageBoardEntry Method (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean)

Gets a specific message from a message board based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

C#

```
public MessageBoardData GetMessageBoardEntry(Long MessageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long callerUserId, Boolean isModerated);
```

Visual Basic

```
Public Function GetMessageBoardEntry(ByVal MessageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long, ByVal
isModerated As Boolean) As MessageBoardData
```

Parameters

Parameters	Description
ByVal MessageId As Long	The unique ID of the message.
ByVal objectId As Long	The ID of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.
ByVal callerUserId As Long	The user ID of the member requesting data.
ByVal isModerated As Boolean	The moderation switch.

Returns

The message board data object with the data from the database.

MessageBoard.IsModerated Method

Check if the object is moderated or not.

C#

```
public Boolean IsModerated(Long objectId, EkEnumeration.MessageBoardObjectType objectType);
```

Visual Basic

```
Public Function IsModerated(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType) As Boolean
```

Parameters

Parameters	Description
ByVal objectId As Long	The id of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.

MessageBoard.IsSpam Method

This function evaluates if the new message is spam or not based on the spam filter selected.

C#

```
public Boolean IsSpam(Long objectId, EkEnumeration.MessageBoardObjectType objectType, Long
userId, String messageText, EkEnumeration.MessageBoardSpamControlType spamControlType, int
interval);
```

Visual Basic

```
Public Function IsSpam(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long, ByVal messageText As String, ByVal
spamControlType As EkEnumeration.MessageBoardSpamControlType, ByVal interval As Integer) As
Boolean
```

Parameters

Parameters	Description
ByVal objectId As Long	The id of the object that has the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of object.
ByVal userId As Long	The Id of the user entering the message.
ByVal messageText As String	The text of the message.
ByVal spamControlType As EkEnumeration.MessageBoardSpamControlType	The type of spam filter selected.
ByVal interval As Integer	The time interval for the spam filter.

MessageBoard.IsUserMessageBoardAdmin Method

This function determines if the user has administrative privileges for the message board.

C#

```
public Boolean IsUserMessageBoardAdmin(Long objectId, EkEnumeration.MessageBoardObjectType
objectType, Long userId);
```

Visual Basic

```
Public Function IsUserMessageBoardAdmin(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal userId As Long	The User (see page 950) ID of the user

Returns

Returns true if the user has administrative privileges.

UpdateMessageText Method

This is the overview for the UpdateMessageText method overload.

Description

This method is overloaded.

- UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String) - updates the text of a message based on the message ID, object ID and type, and the new text.
- UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long) - updates the text of a message based on the message ID, object ID and type, the new text and the ID of the user changing the text.
- UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long, Boolean) - updates the text of a message based on the message ID, object ID and type, the new text, the ID of the user changing the text and whether the board is moderated.

Overload List

	Name	Description
	MessageBoard.UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String) (see page 803)	Updates the Text of a message.
	MessageBoard.UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long) (see page 804)	Updates the Text of a message.
	MessageBoard.UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long, Boolean) (see page 804)	Updates the Text of a message.

MessageBoard.UpdateMessageText Method (Long, Long, EkEnumeration.MessageBoardObjectType, String)

Updates the Text of a message.

C#

```
public MessageBoardData UpdateMessageText(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, String messageText);
```

Visual Basic

```
Public Function UpdateMessageText(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal messageText As String) As
MessageBoardData
```

Parameters

Parameters	Description
ByVal messageId As Long	The unique ID of the message.
ByVal objectId As Long	The ID of the object associated with the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of the object associated with the message board.
ByVal messageText As String	The new message text.

Returns

Returns the updated row.

MessageBoard.UpdateMessageText Method (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long)

Updates the Text of a message.

C#

```
public MessageBoardData UpdateMessageText(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, String messageText, Long callerUserId);
```

Visual Basic

```
Public Function UpdateMessageText(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal messageText As String, ByVal
callerUserId As Long) As MessageBoardData
```

Parameters

Parameters	Description
ByVal messageId As Long	The unique ID of the message.
ByVal objectId As Long	The ID of the object associated with the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of the object associated with the message board.
ByVal messageText As String	The new message text.
ByVal callerUserId As Long	The user ID of the member performing the update.

Returns

Returns the updated row.

Description

'''

MessageBoard.UpdateMessageText Method (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long, Boolean)

Updates the Text of a message.

C#

```
public MessageBoardData UpdateMessageText(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, String messageText, Long callerUserId,
Boolean isModerated);
```

Visual Basic

```
Public Function UpdateMessageText(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal messageText As String, ByVal
callerUserId As Long, ByVal isModerated As Boolean) As MessageBoardData
```

Parameters

Parameters	Description
ByVal messageId As Long	The unique ID of the message.
ByVal objectId As Long	The ID of the object associated with the message board.
ByVal objectType As EkEnumeration.MessageBoardObjectType	The type of the object associated with the message board.
ByVal messageText As String	The new message text.
ByVal callerUserId As Long	The user ID of the member doing the update.
ByVal isModerated As Boolean	The moderation switch.

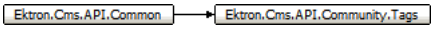
Returns

Returns the updated row.

Tags Class

The Tags class allows you to manage tags for content, users, and community groups. For example, you can add a tag to the system (AddTag (see page 806)), add a tag to a user (AddTagToUser (see page 811)), get a list of tags associated with a user (GetTagsForUser (see page 820)) or delete a tag by its ID (DeleteTagById (see page 812)).

Class Hierarchy



C#

```
public class Tags : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Tags
Inherits Ektron.Cms.API.Common
```

File

Tags.vb

Tags Methods

The methods of the Tags class are listed here.

Public Methods

	Name	Description
	AddTag (see page 806)	Saves a tag and returns an ID. If the tag already exists, the existing tag's ID is returned.
	AddTagToCommunityGroup (see page 806)	This is the overview for the AddTagToCommunityGroup method overload.
	AddTagToContent (see page 808)	This is the overview for the AddTagToContent method overload.
	AddTagToObject (see page 809)	This is the overview for the AddTagToObject method overload.
	AddTagToUser (see page 811)	This is the overview for the AddTagToUser method overload.
	DeleteTagById (see page 812)	Permanently deletes tag and all associations. Even deletes system tags so this should only be called if user has permission to delete system tags. If deleting tags used by a specific user, use DeleteTagsByUser (see page 813).
	DeleteTagOnObject (see page 813)	Deletes a user's tag applied to an object. For example, User (see page 950) **A** has tagged content ID 101 with "CMS" and he now wishes to delete that object tag.
	DeleteTagsByUser (see page 813)	Deletes all tags applied by a user. To delete tags applied to a user, use DeleteTagsOnUser (see page 814).
	DeleteTagsOnCommunityGroup (see page 813)	Deletes all tags applied to a community group.
	DeleteTagsOnContent (see page 814)	Deletes all tags applied to a piece of content.
	DeleteTagsOnObject (see page 814)	Deletes all tags applied to an object.
	DeleteTagsOnUser (see page 814)	Deletes all tags applied to a user. To delete tags applied by a given user, use DeleteTagsByUser (see page 813).
	EditTag (see page 814)	Updates an existing Tag.
	GetAllTagRelatedContent (see page 815)	Returns a list of content related to the supplied content via a common tag. For example, both Content (see page 822) One and Content (see page 822) Two are both tagged with "CMS" and are considered related.
	GetAllTagRelatedUsers (see page 815)	Returns a list of users related to the supplied user via a common tag. For example, both User (see page 950) One and User (see page 950) Two are both tagged with "Engineer" and are considered related.
	GetAllTags (see page 815)	This is the overview for the GetAllTags method overload.

	GetDefaultTags (see page 816)	This is the overview for the GetDefaultTags method overload.
	GetTagByID (see page 818)	Returns a specific tag by providing it's ID.
	GetTagByName (see page 818)	Returns Tag with specified text.
	GetTagCloud (see page 818)	Returns a tag cloud based upon the request.
	GetTagsForContent (see page 818)	Returns an array of tags for a piece of content.
	GetTagsForObject (see page 819)	This is the overview for the GetTagsForObject method overload.
	GetTagsForUser (see page 820)	This is the overview for the GetTagsForUser method overload.
	RemoveTagAsDefault (see page 821)	Removes a tag from the object's default tag list.
	SaveTagAsDefault (see page 821)	Saves a tag to the the object's default list.

Tags.AddTag Method

Saves a tag and returns an ID. If the tag already exists, the existing tag's ID is returned.

C#

```
public Long AddTag(ref TagData tag);
```

Visual Basic

```
Public Function AddTag(ByRef tag As TagData) As Long
```

Parameters

Parameters	Description
ByRef tag As TagData	Tag data to save.

Returns

The ID of newly created tag or existing tag.

AddTagToCommunityGroup Method

This is the overview for the AddTagToCommunityGroup method overload.

Description

This method is overloaded.

- AddTagToCommunityGroup (String, Long) - applies a tag to a community group by providing the tag's text and the community group's ID.
- AddTagToCommunityGroup (String, Long, Integer) - applies a tag in a specified language to a community group by providing the tag's text, the community group's ID, and the language ID.
- AddTagToCommunityGroup (String, Long, Long, Integer) - applies a user created tag in a specified language to a community group by providing the tag's text, the community group's ID, the user ID and the language ID.

Overload List

	Name	Description
	Tags.AddTagToCommunityGroup (String, Long) (see page 807)	Applies a tag to a community group. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language.

	Tags.AddTagToCommunityGroup (String, Long, Integer) (see page 807)	Applies a tag to a community group. If the tag text does not already exist, it is created. Uses the currently logged in user. The newly created ID is accessible via the returned TagData.
	Tags.AddTagToCommunityGroup (String, Long, Long, Integer) (see page 807)	Applies a tag to a community group. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

Tags.AddTagToCommunityGroup Method (String, Long)

Applies a tag to a community group. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language.

C#

```
public TagData AddTagToCommunityGroup(String tagText, Long groupId);
```

Visual Basic

```
Public Function AddTagToCommunityGroup(ByVal tagText As String, ByVal groupId As Long) As
TagData
```

Parameters

Parameters	Description
ByVal tagText As String	Text of tag to apply to object.
ByVal groupId As Long	ID of the community that is being tagged.

Returns

TagData structure representing the applied tag.

Tags.AddTagToCommunityGroup Method (String, Long, Integer)

Applies a tag to a community group. If the tag text does not already exist, it is created. Uses the currently logged in user. The newly created ID is accessible via the returned TagData.

C#

```
public TagData AddTagToCommunityGroup(String tagText, Long groupId, int languageId);
```

Visual Basic

```
Public Function AddTagToCommunityGroup(ByVal tagText As String, ByVal groupId As Long, ByVal
languageId As Integer) As TagData
```

Parameters

Parameters	Description
ByVal tagText As String	Text of tag to apply to object.
ByVal groupId As Long	ID of the community group that is being tagged.
ByVal languageId As Integer	ID of the language the tagText should be saved as. Use -1 for all languages.

Returns

TagData structure representing the applied tag.

Tags.AddTagToCommunityGroup Method (String, Long, Long, Integer)

Applies a tag to a community group. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

C#

```
public TagData AddTagToCommunityGroup(String tagText, Long groupId, Long userId, int
```

```
languageId);
```

Visual Basic

```
Public Function AddTagToCommunityGroup(ByVal tagText As String, ByVal groupId As Long, ByVal
userId As Long, ByVal languageId As Integer) As TagData
```

Parameters

Parameters	Description
ByVal tagText As String	Text of the tag to apply to an object.
ByVal groupId As Long	ID of the community that is being tagged.
ByVal userId As Long	ID of the user applying the tag.
ByVal languageId As Integer	ID of the language the tagText should be saved as. Use -1 for all languages.

Returns

TagData structure representing the applied tag.

AddTagToContent Method

This is the overview for the AddTagToContent method overload.

Description

This method is overloaded.

- AddTagToContent (String, Long) - applies a tag to a piece of content based on the tag's text and the content's ID.
- AddTagToContent (String, Long, Long, Integer) - applies a user created tag in a specified language to a content item by providing the tag's text, the content's ID, the user ID and the language ID.

Overload List

	Name	Description
	Tags.AddTagToContent (String, Long) (see page 808)	Applies a tag to a piece of content. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the tag is saved under the current site language.
	Tags.AddTagToContent (String, Long, Long, Integer) (see page 809)	Applies a tag to a piece of content. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

Tags.AddTagToContent Method (String, Long)

Applies a tag to a piece of content. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the tag is saved under the current site language.

C#

```
public TagData AddTagToContent(String tagText, Long contentId);
```

Visual Basic

```
Public Function AddTagToContent(ByVal tagText As String, ByVal contentId As Long) As TagData
```

Parameters

Parameters	Description
ByVal tagText As String	Text of the tag to apply to an object.
ByVal contentId As Long	ID of the piece of content that is being tagged.

Returns

TagData structure representing the applied tag.

Tags.AddTagToContent Method (String, Long, Long, Integer)

Applies a tag to a piece of content. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

C#

```
public TagData AddTagToContent(String tagText, Long contentId, Long userId, int languageId);
```

Visual Basic

```
Public Function AddTagToContent(ByVal tagText As String, ByVal contentId As Long, ByVal userId
As Long, ByVal languageId As Integer) As TagData
```

Parameters

Parameters	Description
ByVal tagText As String	Text of tag to apply to object.
ByVal contentId As Long	ID of the piece of content that is being tagged.
ByVal userId As Long	ID of the user applying the tag.
ByVal languageId As Integer	ID of the language the tagText should be saved as. Use -1 for all languages.

Returns

TagData structure representing the applied tag.

AddTagToObject Method

This is the overview for the AddTagToObject method overload.

Description

This method is overloaded.

- AddTagToObject (String, Long, EkEnumeration.CMSObjectTypes) - applies a tag to an object based on the tag's text and the object's type and ID.
- AddTagToObject (String, Long, EkEnumeration.CMSObjectTypes, Integer, Integer) - applies a user created tag in a specified language to an object by providing the tag's text, the object's type and ID, and the language ID.

Overload List

	Name	Description
	Tags.AddTagToObject (Long, Long, EkEnumeration.CMSObjectTypes, Long, Integer) (see page 810)	Applies a tag to an object.
	Tags.AddTagToObject (String, Long, EkEnumeration.CMSObjectTypes) (see page 810)	Applies a tag to an object. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language.
	Tags.AddTagToObject (String, Long, EkEnumeration.CMSObjectTypes, Long, Integer) (see page 810)	Applies a tag to an object. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

Tags.AddTagToObject Method (Long, Long, EkEnumeration.CMSObjectTypes, Long, Integer)

Applies a tag to an object.

C#

```
public AddTagToObject(Long tagId, Long objectId, EkEnumeration.CMSObjectTypes objectType, Long
userId, int languageId);
```

Visual Basic

```
Public Sub AddTagToObject(ByVal tagId As Long, ByVal objectId As Long, ByVal objectType As
EkEnumeration.CMSObjectTypes, ByVal userId As Long, ByVal languageId As Integer)
```

Parameters

Parameters	Description
ByVal tagId As Long	Id of the tag to apply to an object.
ByVal objectId As Long	ID of object that is being tagged.
ByVal objectType As EkEnumeration.CMSObjectTypes	Type of object represented by objectId.
ByVal userId As Long	ID of the user applying the tag.
ByVal languageId As Integer	ID of the language the tagText should be saved as. Use -1 for all languages.

Tags.AddTagToObject Method (String, Long, EkEnumeration.CMSObjectTypes)

Applies a tag to an object. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language.

C#

```
public TagData AddTagToObject(String tagText, Long objectId, EkEnumeration.CMSObjectTypes
objectType);
```

Visual Basic

```
Public Function AddTagToObject(ByVal tagText As String, ByVal objectId As Long, ByVal
objectType As EkEnumeration.CMSObjectTypes) As TagData
```

Parameters

Parameters	Description
ByVal tagText As String	Text of the tag to apply to an object.
ByVal objectId As Long	ID of object that is being tagged.
ByVal objectType As EkEnumeration.CMSObjectTypes	Type of object represented by objectId.

Returns

TagData structure representing the applied tag.

Tags.AddTagToObject Method (String, Long, EkEnumeration.CMSObjectTypes, Long, Integer)

Applies a tag to an object. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

C#

```
public TagData AddTagToObject(String tagText, Long objectId, EkEnumeration.CMSObjectTypes
objectType, Long userId, int languageId);
```

Visual Basic

```
Public Function AddTagToObject(ByVal tagText As String, ByVal objectId As Long, ByVal
objectType As EkEnumeration.CMSObjectTypes, ByVal userId As Long, ByVal languageId As Integer)
As TagData
```

Parameters

Parameters	Description
ByVal tagText As String	Text of the tag to apply to an object.
ByVal objectId As Long	ID of object that is being tagged.
ByVal objectType As EkEnumeration.CMSObjectTypes	Type of object represented by objectId.
ByVal userId As Long	ID of the user applying the tag.
ByVal languageId As Integer	ID of the language the tagText should be saved as. Use -1 for all languages.

Returns

TagData structure representing the applied tag.

AddTagToUser Method

This is the overview for the AddTagToUser method overload.

Description

This method is overloaded.

- AddTagToUser (Long, Long) - applies a tag to a user based on a tag's ID and the user's ID. Use this method when a tag already exists.
- AddTagToUser (String, Long) - applies a tag to a user based on the tag's text and the user's ID. Use this method when you are creating a new tag for a user.
- AddTagToUser (String, Long, Integer) - applies a tag in a specified language to a user by providing the tag's text, the user ID and the language ID.

Overload List

	Name	Description
	Tags.AddTagToUser (Long, Long) (see page 811)	Applies a Tag to a user based on a tag's ID and the user's ID.
	Tags.AddTagToUser (String, Long) (see page 812)	Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. The Tag is saved under the current site language.
	Tags.AddTagToUser (String, Long, Integer) (see page 812)	Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

Tags.AddTagToUser Method (Long, Long)

Applies a Tag to a user based on a tag's ID and the user's ID.

C#

```
public AddTagToUser(Long tagId, Long userId);
```

Visual Basic

```
Public Sub AddTagToUser(ByVal tagId As Long, ByVal userId As Long)
```

Parameters

Parameters	Description
ByVal tagId As Long	ID of tag to apply to object.
ByVal userId As Long	ID of the user that is being tagged.

Tags.AddTagToUser Method (String, Long)

Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. The Tag is saved under the current site language.

C#

```
public TagData AddTagToUser(String tagText, Long userId);
```

Visual Basic

```
Public Function AddTagToUser(ByVal tagText As String, ByVal userId As Long) As TagData
```

Parameters

Parameters	Description
ByVal tagText As String	Text of the tag to apply to an object.
ByVal userId As Long	ID of the user that is being tagged.

Returns

TagData structure representing the applied tag.

Tags.AddTagToUser Method (String, Long, Integer)

Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

C#

```
public TagData AddTagToUser(String tagText, Long userId, int languageId);
```

Visual Basic

```
Public Function AddTagToUser(ByVal tagText As String, ByVal userId As Long, ByVal languageId
As Integer) As TagData
```

Parameters

Parameters	Description
ByVal tagText As String	Text of the tag to apply to an object.
ByVal userId As Long	ID of the user that is being tagged.
ByVal languageId As Integer	ID of language the tag should be saved as.

Returns

TagData structure representing the applied tag.

Tags.DeleteTagById Method

Permanently deletes tag and all associations. Even deletes system tags so this should only be called if user has permission to delete system tags. If deleting tags used by a specific user, use DeleteTagsByUser (see page 813).

C#

```
public Boolean DeleteTagById(Long tagId);
```

Visual Basic

```
Public Function DeleteTagById(ByVal tagId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal tagId As Long	ID of tag to be deleted.

Tags.DeleteTagOnObject Method

Deletes a user's tag applied to an object. For example, User (see page 950) **A** has tagged content ID 101 with "CMS" and he now wishes to delete that object tag.

C#

```
public Boolean DeleteTagOnObject(Long tagID, Long ObjectID, EkEnumeration.CMSObjectTypes
ObjectType, Long userId);
```

Visual Basic

```
Public Function DeleteTagOnObject(ByVal tagID As Long, ByVal ObjectID As Long, ByVal
ObjectType As EkEnumeration.CMSObjectTypes, ByVal userId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal tagID As Long	ID of tag to delete.
ByVal ObjectID As Long	ID of the object to delete tags.
ByVal ObjectType As EkEnumeration.CMSObjectTypes	The type of object associated with the ObjectId.
ByVal userId As Long	The user who originally applied the tag and wants to delete it.

Returns

True if operation was successful.

Tags.DeleteTagsByUser Method

Deletes all tags applied by a user. To delete tags applied to a user, use DeleteTagsOnUser (see page 814).

C#

```
public Boolean DeleteTagsByUser(Long userId);
```

Visual Basic

```
Public Function DeleteTagsByUser(ByVal userId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal userId As Long	ID of user whose tags are being deleted.

Returns

True if operation was successful.

Tags.DeleteTagsOnCommunityGroup Method

Deletes all tags applied to a community group.

C#

```
public Boolean DeleteTagsOnCommunityGroup(Long groupId);
```

Visual Basic

```
Public Function DeleteTagsOnCommunityGroup(ByVal groupId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal groupId As Long	ID of community group that has associated tags you wish to delete.

Returns

True if operation was successful.

Tags.DeleteTagsOnContent Method

Deletes all tags applied to a piece of content.

C#

```
public Boolean DeleteTagsOnContent(Long contentId);
```

Visual Basic

```
Public Function DeleteTagsOnContent(ByVal contentId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal contentId As Long	ID of a content item to delete associated tags.

Returns

True if operation was successful.

Tags.DeleteTagsOnObject Method

Deletes all tags applied to an object.

C#

```
public Boolean DeleteTagsOnObject(Long objectId, EkEnumeration.CMSObjectTypes objectType);
```

Visual Basic

```
Public Function DeleteTagsOnObject(ByVal objectId As Long, ByVal objectType As
EkEnumeration.CMSObjectTypes) As Boolean
```

Parameters

Parameters	Description
ByVal objectId As Long	ID of the object to delete associated tags.
ByVal objectType As EkEnumeration.CMSObjectTypes	The type of object associated with the ObjectId.

Returns

True if operation was successful.

Tags.DeleteTagsOnUser Method

Deletes all tags applied to a user. To delete tags applied by a given user, use DeleteTagsByUser (see page 813).

C#

```
public Boolean DeleteTagsOnUser(Long userId);
```

Visual Basic

```
Public Function DeleteTagsOnUser(ByVal userId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal userId As Long	ID of the user whose associated tags are being deleted.

Returns

True if operation was successful.

Tags.EditTag Method

Updates an existing Tag.

C#

```
public TagData EditTag(ref TagData tag);
```

Visual Basic

```
Public Function EditTag(ByRef tag As TagData) As TagData
```

Parameters

Parameters	Description
ByRef tag As TagData	Tag data to save.

Returns

ID of newly created tag or existing tag.

Tags.GetAllTagRelatedContent Method

Returns a list of content related to the supplied content via a common tag. For example, both Content (see page 822) One and Content (see page 822) Two are both tagged with "CMS" and are considered related.

C#

```
public ContentData[] GetAllTagRelatedContent(Long contentId);
```

Visual Basic

```
Public Function GetAllTagRelatedContent(ByVal contentId As Long) As ContentData()
```

Parameters

Parameters	Description
ByVal contentId As Long	This is the piece of content for which to retrieve related content.

Returns

Returns array of partially populated ContentData objects.

Tags.GetAllTagRelatedUsers Method

Returns a list of users related to the supplied user via a common tag. For example, both User (see page 950) One and User (see page 950) Two are both tagged with "Engineer" and are considered related.

C#

```
public UserData[] GetAllTagRelatedUsers(Long userId);
```

Visual Basic

```
Public Function GetAllTagRelatedUsers(ByVal userId As Long) As UserData()
```

Parameters

Parameters	Description
ByVal userId As Long	This is the user for which to retrieve related users.

Returns

Array of partially populated UserData objects.

GetAllTags Method

This is the overview for the GetAllTags method overload.

Description

This method is overloaded.

- GetAllTags (PagingInfo) - retrieves all tags in the system. This method allows for paging.
- GetAllTags (TagRequestData, Integer) - retrieves a list of tags in the system based on the requested information and the total number of tags requested.

Overload List

	Name	Description
	Tags.GetAllTags (PagingInfo) (see page 816)	Returns a list of all tags in the system.
	Tags.GetAllTags (TagRequestData, Integer) (see page 816)	Returns a list of all tags in the system.

Tags.GetAllTags Method (PagingInfo)

Returns a list of all tags in the system.

C#

```
public TagData[] GetAllTags(PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetAllTags(ByVal pagingInfo As PagingInfo) As TagData()
```

Parameters

Parameters	Description
ByVal pagingInfo As PagingInfo	Paging information for call.

Returns

An array of TagData objects.

Tags.GetAllTags Method (TagRequestData, Integer)

Returns a list of all tags in the system.

C#

```
public TagData[] GetAllTags(TagRequestData request, ref int totalTags);
```

Visual Basic

```
Public Function GetAllTags(ByVal request As TagRequestData, ByRef totalTags As Integer) As
TagData()
```

Parameters

Parameters	Description
ByVal request As TagRequestData	Contains the parameters for the tags to retrieve.
ByRef totalTags As Integer	Returns the total number of tags available for the requested parameters.

Returns

Array of TagData objects

GetDefaultTags Method

This is the overview for the GetDefaultTags method overload.

Description

This method is overloaded.

- GetDefaultTags (CMSObjectTypes) - gets default tags for a specified object.
- GetDefaultTags (CMSObjectTypes, Integer) - gets default tags in a specified language for an object.

Overload List

	Name	Description
	Tags.GetDefaultTags (CMSObjectTypes) (see page 817)	Returns a list of all system defined tags in the system.
	Tags.GetDefaultTags (CMSObjectTypes, Integer) (see page 817)	Returns a list of all system defined tags in the system.

Tags.GetDefaultTags Method (CMSObjectTypes)

Returns a list of all system defined tags in the system.

C#

```
public TagData[] GetDefaultTags(CMSObjectTypes objectType);
```

Visual Basic

```
Public Function GetDefaultTags(ByVal objectType As CMSObjectTypes) As TagData()
```

Parameters

Parameters	Description
ByVal objectType As CMSObjectTypes	The type of object for which to retrieve tags. For example, retrieve all the default tags for CommunityGroups (CMSObjectTypes.CommunityGroup (see page 750)).

Returns

Collection (see page 95) of TagData objects

Tags.GetDefaultTags Method (CMSObjectTypes, Integer)

Returns a list of all system defined tags in the system.

C#

```
public TagData[] GetDefaultTags(CMSObjectTypes objectType, int languageId);
```

Visual Basic

```
Public Function GetDefaultTags(ByVal objectType As CMSObjectTypes, ByVal languageId As
Integer) As TagData()
```

Parameters

Parameters	Description
ByVal objectType As CMSObjectTypes	The type of object for which to retrieve tags. For example, retrieve all the default tags for CommunityGroups (CMSObjectTypes.CommunityGroup (see page 750)).
ByVal languageId As Integer	Language of tags to retrieve.

Returns

Collection (see page 95) of TagData objects

Tags.GetTagByID Method

Returns a specific tag by providing it's ID.

C#

```
public TagData GetTagByID(Long tagId);
```

Visual Basic

```
Public Function GetTagByID(ByVal tagId As Long) As TagData
```

Parameters

Parameters	Description
ByVal tagId As Long	ID of the tag to retrieve.

Returns

Returns a tag found with supplied ID or new tag if no tag is found.

Tags.GetTagByName Method

Returns Tag with specified text.

C#

```
public TagData GetTagByName(String tagText);
```

Visual Basic

```
Public Function GetTagByName(ByVal tagText As String) As TagData
```

Parameters

Parameters	Description
ByVal tagText As String	Text of the tag to retrieve.

Returns

Returns tag found with supplied ID or new tag if no tag is found.

Tags.GetTagCloud Method

Returns a tag cloud based upon the request.

C#

```
public TagCloud GetTagCloud(TagCloudRequestData request, ref int totalTags);
```

Visual Basic

```
Public Function GetTagCloud(ByVal request As TagCloudRequestData, ByRef totalTags As Integer)
As TagCloud
```

Parameters

Parameters	Description
ByVal request As TagCloudRequestData	Contains the parameters for the tag cloud to retrieve.
ByRef totalTags As Integer	Returns the total number of tags available in the requested cloud.

Returns

A TagCloud (see page 513) object representing the request.

Tags.GetTagsForContent Method

Returns an array of tags for a piece of content.

C#

```
public TagData[] GetTagsForContent(Long contentId);
```

Visual Basic

```
Public Function GetTagsForContent(ByVal contentId As Long) As TagData()
```

Parameters

Parameters	Description
ByVal contentId As Long	The ID of the piece of content for which to retrieve tags.

Returns

An array of TagData objects that have been applied to the content.

GetTagsForObject Method

This is the overview for the GetTagsForObject method overload.

Description

This method is overloaded.

- GetTagsForObject (Long, CMSObjectTypes, Integer) - returns an array of tags for a given object based on the object's ID and type, and the language ID.
- GetTagsForObject (Long, EkEnumeration.CMSObjectTypes) - returns an array of tags for a given object based on the object's ID and type.

Overload List

	Name	Description
	Tags.GetTagsForObject (Long, CMSObjectTypes, Integer) (see page 819)	Returns an array of tags for a given object.
	Tags.GetTagsForObject (Long, EkEnumeration.CMSObjectTypes) (see page 820)	Returns an array of tags for a given object.

Tags.GetTagsForObject Method (Long, CMSObjectTypes, Integer)

Returns an array of tags for a given object.

C#

```
public TagData[] GetTagsForObject(Long objectId, CMSObjectTypes objectType, int languageId);
```

Visual Basic

```
Public Function GetTagsForObject(ByVal objectId As Long, ByVal objectType As CMSObjectTypes,
ByVal languageId As Integer) As TagData()
```

Parameters

Parameters	Description
ByVal objectId As Long	The object ID to retrieve tags for.
ByVal objectType As CMSObjectTypes	The object type to retrieve tags for.
ByVal languageId As Integer	The language ID of the tags to retrieve.

Returns

An array of TagData objects that have been applied to an object.

Tags.GetTagsForObject Method (Long, EkEnumeration.CMSObjectTypes)

Returns an array of tags for a given object.

C#

```
public TagData[] GetTagsForObject(Long objectId, EkEnumeration.CMSObjectTypes objectType);
```

Visual Basic

```
Public Function GetTagsForObject(ByVal objectId As Long, ByVal objectType As
EkEnumeration.CMSObjectTypes) As TagData()
```

Parameters

Parameters	Description
ByVal objectId As Long	The ID of the object to retrieve tags for.
ByVal objectType As EkEnumeration.CMSObjectTypes	The Object type to retrieve tags for.

Returns

An array of TagData objects that have been applied to an object.

GetTagsForUser Method

This is the overview for the GetTagsForUser method overload.

Description

This method is overloaded.

- GetTagsForUser (Long) - returns an array of tags for a user based on the user's ID.
- GetTagsForUser (Long, Integer) - returns an array of tags for a specific language associated with a user based on the user's ID and language ID.

Overload List

	Name	Description
	Tags.GetTagsForUser (Long) (see page 820)	Returns an array of tags for a user.
	Tags.GetTagsForUser (Long, Integer) (see page 821)	Returns an array of tags for a user.

Tags.GetTagsForUser Method (Long)

Returns an array of tags for a user.

C#

```
public TagData[] GetTagsForUser(Long userId);
```

Visual Basic

```
Public Function GetTagsForUser(ByVal userId As Long) As TagData()
```

Parameters

Parameters	Description
ByVal userId As Long	The ID of the user to retrieve tags for.

Returns

An array of TagData objects that have been applied to the user.

Tags.GetTagsForUser Method (Long, Integer)

Returns an array of tags for a user.

C#

```
public TagData[] GetTagsForUser(Long userId, int languageId);
```

Visual Basic

```
Public Function GetTagsForUser(ByVal userId As Long, ByVal languageId As Integer) As TagData()
```

Parameters

Parameters	Description
ByVal userId As Long	ID of the user to retrieve tags for.
ByVal languageId As Integer	Language of tags to retrieve.

Returns

An array of TagData objects that have been applied to the user.

Tags.RemoveTagAsDefault Method

Removes a tag from the object's default tag list.

C#

```
public RemoveTagAsDefault(String tagText, int languageId, EkEnumeration.CMSObjectTypes
objectType);
```

Visual Basic

```
Public Sub RemoveTagAsDefault(ByVal tagText As String, ByVal languageId As Integer, ByVal
objectType As EkEnumeration.CMSObjectTypes)
```

Parameters

Parameters	Description
ByVal tagText As String	The text of the tag.
ByVal languageId As Integer	The ID of the tag being removed.
ByVal objectType As EkEnumeration.CMSObjectTypes	The type of object.

Tags.SaveTagAsDefault Method

Saves a tag to the the object's default list.

C#

```
public TagData SaveTagAsDefault(String tagText, int languageId, EkEnumeration.CMSObjectTypes
objectType);
```

Visual Basic

```
Public Function SaveTagAsDefault(ByVal tagText As String, ByVal languageId As Integer, ByVal
objectType As EkEnumeration.CMSObjectTypes) As TagData
```

Parameters

Parameters	Description
ByVal tagText As String	The text of the tag.
ByVal languageId As Integer	The ID of the language to which the tag will be assigned.
ByVal objectType As EkEnumeration.CMSObjectTypes	The type of object.

Ektron.Cms.API.Content Namespace

This is namespace Ektron.Cms.API.Content.

Classes

	Name	Description
	Asset (see page 822)	This is class Ektron.Cms.API.Content.Asset.
	Blog (see page 826)	This is class Ektron.Cms.API.Content.Blog.
	Content (see page 840)	This is class Ektron.Cms.API.Content.Content.
	ContentRating (see page 866)	This is class Ektron.Cms.API.Content.ContentRating.
	Form (see page 871)	This is class Ektron.Cms.API.Content.Form.
	Taxonomy (see page 876)	This is class Ektron.Cms.API.Content.Taxonomy.
	ThreadedDiscussion (see page 882)	The ThreadedDiscussion class allows you to work with threaded discussion by adding and removing discussion boards, forums, and topics. It also allows you to subscribe or unsubscribe to a board, forum or topic. In addition, you can retrieve a specific topic or search a topic's replies. The Threaded Discussion feature provides a discussion board where a forum can be created and topics can be discussed on your Web site. A site or membership user with Add Topic permission creates a topic and posts a starter question. Other site visitors can then reply to the topic.

Classes

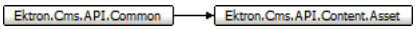
The following table lists classes in this documentation.

Classes

	Name	Description
	Asset (see page 822)	This is class Ektron.Cms.API.Content.Asset.
	Blog (see page 826)	This is class Ektron.Cms.API.Content.Blog.
	Content (see page 840)	This is class Ektron.Cms.API.Content.Content.
	ContentRating (see page 866)	This is class Ektron.Cms.API.Content.ContentRating.
	Form (see page 871)	This is class Ektron.Cms.API.Content.Form.
	Taxonomy (see page 876)	This is class Ektron.Cms.API.Content.Taxonomy.
	ThreadedDiscussion (see page 882)	The ThreadedDiscussion class allows you to work with threaded discussion by adding and removing discussion boards, forums, and topics. It also allows you to subscribe or unsubscribe to a board, forum or topic. In addition, you can retrieve a specific topic or search a topic's replies. The Threaded Discussion feature provides a discussion board where a forum can be created and topics can be discussed on your Web site. A site or membership user with Add Topic permission creates a topic and posts a starter question. Other site visitors can then reply to the topic.

Asset Class

Class Hierarchy



C#

```
public class Asset : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Asset
Inherits Ektron.Cms.API.Common
```

File

Asset.vb

Description

This is class Ektron.Cms.API.Content.Asset.

Asset Methods

The methods of the Asset class are listed here.

Public Methods

	Name	Description
	GetAssetMgtConfigInfo (see page 823)	Loads all of the registered plug-in configuration details. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)
	GetAssetMgtInfo (see page 823)	Loads all of the registered plug-in type details. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)
	GetAssetSupertypes (see page 824)	Loads all of the asset supertypes supported by the documents add-in. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)
	GetEditSnippet (see page 824)	Loads the edit snippet for the given document. Used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)
	GetMyAssetMgtID (see page 824)	Retrieves the Document Management functionality client ID. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)
	GetViewUrl (see page 825)	Returns a URL associated with the current asset. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)
	MakeCopyDMSAsset (see page 825)	Create a copy of a DMS Asset (see page 822). This method is a member of Ektron.Cms.API.Content.Asset (see page 822)
	SetAssetMgtConfigInfo (see page 825)	Updates the Document Management configuration information. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)

Asset.GetAssetMgtConfigInfo Method

Loads all of the registered plug-in configuration details. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)

C#

```
public AssetConfigInfo[] GetAssetMgtConfigInfo();
```

Visual Basic

```
Public Function GetAssetMgtConfigInfo() As AssetConfigInfo()
```

Returns

Ektron.Cms.AssetConfigInfo()

Asset.GetAssetMgtInfo Method

Loads all of the registered plug-in type details. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)

C#

```
public AssetInfoData[] GetAssetMgtInfo();
```

Visual Basic

```
Public Function GetAssetMgtInfo() As AssetInfoData()
```

Returns

Ektron.Cms.AssetInfoData()

Asset.GetAssetSupertypes Method

Loads all of the asset supertypes supported by the documents add-in. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)

C#

```
public AssetInfoData[] GetAssetSupertypes();
```

Visual Basic

```
Public Function GetAssetSupertypes() As AssetInfoData()
```

Returns

Ektron.Cms.AssetInfoData()

Asset.GetEditSnippet Method

Loads the edit snippet for the given document. Used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)

C#

```
public String GetEditSnippet(String AssetId, int ContentType);
```

Visual Basic

```
Public Function GetEditSnippet(ByVal AssetId As String, ByVal ContentType As Integer) As String
```

Parameters

Parameters	Description
ByVal AssetId As String	The document's ID (Required)
ByVal ContentType As Integer	The type of the content, such as office documents or managed files or * (all).

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetMyAssetMgtID Method

Retrieves the Document Management functionality client ID. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)

C#

```
public String GetMyAssetMgtID();
```

Visual Basic

```
Public Function GetMyAssetMgtID() As String
```

Returns

String

Asset.GetViewUrl Method

Returns a URL associated with the current asset. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)

C#

```
public String GetViewUrl(String AssetId, int ContentType);
```

Visual Basic

```
Public Function GetViewUrl(ByVal AssetId As String, ByVal ContentType As Integer) As String
```

Parameters

Parameters	Description
ByVal AssetId As String	The document's ID (Required)
ByVal ContentType As Integer	The type of the content, such as office documents or managed files or * (all).

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.MakeCopyDMSAsset Method

Create a copy of a DMS Asset (see page 822). This method is a member of Ektron.Cms.API.Content.Asset (see page 822)

C#

```
public Boolean MakeCopyDMSAsset(ref String strAssetID, ref String strAssetVersion);
```

Visual Basic

```
Public Function MakeCopyDMSAsset(ByRef strAssetID As String, ByRef strAssetVersion As String)
As Boolean
```

Parameters

Parameters	Description
ByRef strAssetID As String	ID associated with the asset
ByRef strAssetVersion As String	Version of the asset

Returns

boolean

Asset.SetAssetMgtConfigInfo Method

Updates the Document Management configuration information. This method is a member of Ektron.Cms.API.Content.Asset (see page 822)

C#

```
public Boolean SetAssetMgtConfigInfo(AssetConfigInfo asset_config);
```

Visual Basic

```
Public Function SetAssetMgtConfigInfo(ByVal asset_config() As AssetConfigInfo) As Boolean
```

Parameters

Parameters	Description
ByVal asset_config() As AssetConfigInfo	The Document Management configuration data

Returns

Boolean

Blog Class

Class Hierarchy

C#

```
public class Blog : Content;
```

Visual Basic

```
Public Class Blog
Inherits Content
```

File

Blog.vb

Description

This is class Ektron.Cms.API.Content.Blog.

Blog Methods

The methods of the Blog class are listed here.

Public Methods

	Name	Description
	AddBlog (see page 827)	Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.
	AddCommentForPost (see page 827)	Adds a post comment to a blog post. Accepts a BlogComment class as the input. Will return the result of that add action. False indicates the action was successful.
	AddPost (see page 827)	Adds a post to a given blog. Accepts a ContentData class as the post content. Return new post ID. *See Also:* The Example section.
	DeleteBlog (see page 829)	Deletes a blog with a given ID. This will delete the blog as well as all posts and comments for the blog. BlogID must be greater than 0.
	DeleteBlogPostComment (see page 830)	Delete Blog (see page 826) post comment by providing the comment's ID.
	DeletePost (see page 830)	Deletes a post by the post's ID. This method will delete all comments associated with the post as well. Returns boolean value.
	GetBlankBlogPostData (see page 830)	Returns a blank BlogPostData class. Used Primarily in the Workarea.
	GetBlog (see page 830)	This is the overview for the GetBlog method overload.
	GetBlogData (see page 834)	This is the overview for the GetBlogData method overload.
	GetBlogDataForPost (see page 837)	Returns the blog information for a blog post. This contains information about categories, whether comments are enabled, etc. Accepts the post ID.
	GetBlogPostData (see page 837)	Gets the Post data for a specific blog post. Accepts the ID of the post. Will return tags, trackback, pingback, and categories for the post.
	GetBlogRoll (see page 837)	This is the overview for the GetBlogRoll method overload.
	GetPost (see page 839)	Gets a blog post for a given ID. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData Class.
	GetPostbyID (see page 839)	Gets a blog post for a given ID. Includes trackback, pinback, comments, and other information. Returns BlogPostData Class.
	GetUserBlog (see page 839)	gets the user blog id
	UpdateBlogProperties (see page 840)	Updates the properties of a blog. Accepts standard blogdata object. Returns boolean.

Blog.AddBlog Method

Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.

C#

```
public Long AddBlog(Long ParentFolderID, String BlogName, String BlogTitle, String
BlogDescription, Ektron.Cms.Common.EkEnumeration.BlogVisibility Visibility, Boolean
EnableComments, Boolean ModerateComments, Boolean RequireAuthentication, String Categories,
BlogRoll RollLinks);
```

Visual Basic

```
Public Function AddBlog(ByVal ParentFolderID As Long, ByVal BlogName As String, ByVal
BlogTitle As String, ByVal BlogDescription As String, ByVal Visibility As
Ektron.Cms.Common.EkEnumeration.BlogVisibility, ByVal EnableComments As Boolean, ByVal
ModerateComments As Boolean, ByVal RequireAuthentication As Boolean, ByVal Categories() As
String, ByVal RollLinks As BlogRoll) As Long
```

Parameters

Parameters	Description
ByVal ParentFolderID As Long	
ByVal BlogName As String	Name of the Blog (see page 826).
ByVal BlogTitle As String	Title of the Blog (see page 826).
ByVal BlogDescription As String	Description of the Blog (see page 826).
ByVal Visibility As Ektron.Cms.Common.EkEnumeration.BlogVisibility	If the blog is public, enter Public. If private, enter Private.
ByVal EnableComments As Boolean	Enter True to allow commenting, otherwise enter false.
ByVal ModerateComments As Boolean	If commenting is allowed, enter True to moderate comments.
ByVal RequireAuthentication As Boolean	If commenting is allowed, require authentication to add comment.
ByVal Categories() As String	String Array representing the categories for this blog.
ByVal RollLinks As BlogRoll	The Blog (see page 826)'s roll links.

Returns

Numeric value of the blog ID that is greater than 0 (zero).

Blog.AddCommentForPost Method

Adds a post comment to a blog post. Accepts a BlogComment class as the input. Will return the result of that add action. False indicates the action was successful.

C#

```
public Boolean AddCommentForPost(BlogComment bcComment);
```

Visual Basic

```
Public Function AddCommentForPost(ByVal bcComment As BlogComment) As Boolean
```

Parameters

Parameters	Description
ByVal bcComment As BlogComment	A populated BlogComment class. Title, DisplayName, Comment text, Email are required.

Returns

A Boolean value indicating whether there was a problem. False indicates the comment was added successfully.

Blog.AddPost Method

Adds a post to a given blog. Accepts a ContentData class as the post content. Return new post ID.

See Also: The Example section.

C#

```
public Long AddPost(Long blogID, ContentData PostContent, String Categories, Boolean Pingback,
String Tags, String TrackbackURL);
```

Visual Basic

```
Public Function AddPost(ByVal blogID As Long, ByVal PostContent As ContentData, ByVal
Categories() As String, ByVal Pingback As Boolean, ByVal Tags As String, ByVal TrackbackURL As
String) As Long
```

Parameters

Parameters	Description
ByVal blogID As Long	Numeric ID of the blog to which the post is added.
ByVal PostContent As ContentData	ContentData class representing the HTML, title, etc.
ByVal Categories() As String	String Array representing the categories to which this post applies.
ByVal Pingback As Boolean	Pingback URL(s) in this post?
ByVal Tags As String	String list of tags that applies to this post.
ByVal TrackbackURL As String	Trackback URL(s) to notify.

Returns

Numeric ID of the post created. (Long)

Example

The following example shows how to create a Web page from which a user can add a blog post. A user enters the blog's ID, post title and content for the post. This example uses some standard drag and drop controls and a small section of VB code utilizing the AddPost method. This method uses the InternalAdmin to add the blog post. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, create a table with three rows and three columns.
2. In the first row, first column, add the following label for the blog ID.
```
<asp:Label ID="Label6" runat="server" Text="Blog ID:" Width="76px"></asp:Label>
```
3. In the first row, second column, add a text box to enter the blog's ID.
```
<asp:TextBox ID="txtBlogId" runat="server" Width="55px"></asp:TextBox>
```
4. In the first row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the blog ID box.
```
<asp:RequiredFieldValidator ID="rfvBlogId" runat="server" ControlToValidate="txtBlogId"
  ErrorMessage="Please enter user name">Please enter blog ID</asp:RequiredFieldValidator>
```
5. In the second row, first column, add the following label for the area to post content.
```
<asp:Label ID="Label4" runat="server" Text="Post content:" Width="82px"></asp:Label>
```
6. In the second row, second column, add text box where the user can enter the blog post's content.
```
<asp:TextBox ID="txtPostContent" runat="server" Width="144px"></asp:TextBox>
```
7. In the second row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the blog post's content box.
```
<asp:RequiredFieldValidator ID="rfvPostContent" runat="server"
ControlToValidate="txtPostContent"
  ErrorMessage="Please enter first name">Please enter content to
post</asp:RequiredFieldValidator>
```
8. In the third row, first column, add the following label for the blog post's title.
```
<asp:Label ID="Label7" runat="server" Text="Post Title: " Width="76px"></asp:Label>
```
9. In the third row, second column, add text box where the user can enter the blog post's title.
```
<asp:TextBox ID="txtPostTitle" runat="server" Width="144px">Post Title</asp:TextBox>
```
10. In the third row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the blog post's title box.
```
<asp:RequiredFieldValidator ID="rfvPostTitle" runat="server"
ControlToValidate="txtPostTitle"
  ErrorMessage="Please enter first name">Please enter title for the content to
post</asp:RequiredFieldValidator>
```
11. After the table, add a break and then add a button to submit the blog post.
```
<br />
```

```
<asp:Button ID="btnAdd" runat="server" Text="Add Post" Width="70px" />
```

12. Add a some breaks for spacing purposes.

```
<br /><br />
```

13. Add a results label that lets a user know that the blog post has been added and the blog post's ID.

```
<asp:Label ID="lblResult" runat="server" Width="432px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
btnAdd.Click
```

2. Check to make sure the blog ID is numeric.

```
If Not IsNumeric(txtblogId.Text) Then
    lblResult.Text = "Invalid Blog ID"
    Exit Sub
End If
```

3. Create a Try/Catch around the AddPost method. This allows you to catch any exceptions and helps when debugging.
4. Create objects for the following items.

```
Try
   Dim blogApi As New Ektron.Cms.API.Content.Blog
   Dim postContentData As New Ektron.Cms.ContentData
   Dim categories() As String = New String() {}
```

5. Populate postContentData with the Title and the Content (see page 840) of the blog post.

```
postContentData.Title = txtPostTitle.text
postContentData.Html = txtPostContent.Text
```

6. Create a place holder for blog post's ID using the AddPost method.

```
Dim postId As Integer
postId = blogApi.AddPost(CInt(txtBlogId.Text), postContentData, categories, False,
"", "")
```

7. Let the user know the post ID and that it has been added.

```
lblResult.Text = "A post was made to the blog with post ID " & postId
```

8. If there is an error, let the user know what happened.

```
Catch ex As Exception
   lblResult.Text = ex.Message
End Try

End Sub
```

Blog.DeleteBlog Method

Deletes a blog with a given ID. This will delete the blog as well as all posts and comments for the blog. BlogID must be greater than 0.

C#

```
public DeleteBlog(Long BlogID);
```

Visual Basic

```
Public Sub DeleteBlog(ByVal BlogID As Long)
```

Parameters

Parameters	Description
ByVal BlogID As Long	The numeric ID of the blog. This corresponds to the folder ID of the blog.

Blog.DeleteBlogPostComment Method

Delete Blog (see page 826) post comment by providing the comment's ID.

C#

```
public DeleteBlogPostComment(Long postCommentId);
```

Visual Basic

```
Public Sub DeleteBlogPostComment(ByVal postCommentId As Long)
```

Parameters

Parameters	Description
ByVal postCommentId As Long	Post comment ID.

Blog.DeletePost Method

Deletes a post by the post's ID. This method will delete all comments associated with the post as well. Returns boolean value.

C#

```
public Boolean DeletePost(Long post_id);
```

Visual Basic

```
Public Function DeletePost(ByVal post_id As Long) As Boolean
```

Parameters

Parameters	Description
ByVal post_id As Long	The numeric value of the post ID to delete.

Returns

Boolean

Blog.GetBlankBlogPostData Method

Returns a blank BlogPostData class. Used Primarily in the Workarea.

C#

```
public BlogPostData GetBlankBlogPostData();
```

Visual Basic

```
Public Function GetBlankBlogPostData() As BlogPostData
```

Returns

A blank, but not null, BlogPostData class.

GetBlog Method

This is the overview for the GetBlog method overload.

Description

Returns a BlogData class for a given blog, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. This method is overloaded. One GetBlog method allows you to use the Blog's ID. The other, allows you to use a Blog's path.

Overload List

	Name	Description
	Blog.GetBlog (Long, String, Integer, Integer) (see page 831)	Returns a BlogData class for a given blog ID, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.
	Blog.GetBlog (String, String, Integer, Integer) (see page 833)	Returns a BlogData class for a given path, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

Blog.GetBlog Method (Long, String, Integer, Integer)

Returns a BlogData class for a given blog ID, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

C#

```
public BlogData GetBlog(Long ID, String StartDate, int MaxEntries, int LanguageID);
```

Visual Basic

```
Public Function GetBlog(ByVal ID As Long, ByVal StartDate As String, ByVal MaxEntries As
Integer, ByVal LanguageID As Integer) As BlogData
```

Parameters

Parameters	Description
ByVal ID As Long	The ID of the blog. This is a numeric value that corresponds to the folder ID of the blog.
ByVal StartDate As String	The date to start getting entries. Used if you want to get historical information. Gets information only for the date specified.
ByVal MaxEntries As Integer	The total number of posts to return. This is an integer value. If set to 0, the `postsvisible` property of the blog is used. If set to -1, the content is not returned, only the post info for the content.
ByVal LanguageID As Integer	The language ID for the content to get. An integer value.

Returns

For a given blog ID, this returns either null or a BlogData class with the associated posts. This also includes the roll, categories, and post information.

Remarks

Developers are advised to check for null prior to using.

Example

The following example shows how to create a Web page from which a user can retrieve blog information by specifying a blog ID, a date and how to handle post information. The sample returns BlogData for the Title, Tagline, Categories, Blog (see page 826) roll links and the number of posts. To create this example, we use some standard drag and drop controls and a small section of VB code utilizing the GetBlog method. This method uses the InternalAdmin to retrieve the blog information. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Create a table with three rows and two columns.
2. In the first row, first column, add a label that shows where to enter the blog's ID.
```
<asp:Label ID="Label1" runat="server" Text="Blog ID: "></asp:Label>
```
3. In the first row, second column, add a text box for the user to enter the blog ID and a RequiredFieldValidator to validate that something appears in the blog ID box.
```
<asp:TextBox ID="txtBlogId" runat="server" Width="63px">41</asp:TextBox>
<asp:RequiredFieldValidator ID="rfvBlogId" runat="server" ControlToValidate="txtBlogId"
    ErrorMessage="Please enter blog ID"></asp:RequiredFieldValidator>
```
4. In the second row, first column, add a label that shows where to enter the date.
```
<asp:Label ID="Label2" runat="server" Text="Date: "></asp:Label>
```

5. In the second row, second column, add a text box for the user to enter the date and a RequiredFieldValidator to validate that something appears in the date box.

```
<asp:TextBox ID="txtDate" runat="server" Width="63px"></asp:TextBox>
<asp:RequiredFieldValidator ID="rfvDate" runat="server" ControlToValidate="txtDate"
   ErrorMessage="Please enter the post date"></asp:RequiredFieldValidator>
```

6. In the third row, first column, add a label for the posts.

```
<asp:Label ID="Label4" runat="server" Text="Posts:"></asp:Label>
```

7. In the third row, second column, add the following radio buttons with breaks between each one.

```
<asp:RadioButton ID="rb_forday" runat="server" Checked="True" GroupName="rb_num" Text="All
posts for that day only" /><br />
<asp:RadioButton ID="rb_dayprior" runat="server" GroupName="rb_num" Text="All posts for
that day and before" /><br />
<asp:RadioButton ID="rb_month" runat="server" GroupName="rb_num" Text="All posts for the
month that day falls on." /><br />
<asp:RadioButton ID="rb_Custom" runat="server" GroupName="rb_num" Text="Starting from the
date, this number of posts:" />
```

8. Add a text box at the end of the last radio button so a user can enter the quantity of posts to return.

```
<asp:TextBox ID="txt_num" runat="server" Width="63px">1</asp:TextBox>
```

9. After the table, add a break.

```
<br />
```

10. Add a button to retrieve the blog information.

```
<asp:Button ID="btnGetBlog" runat="server" Text="Get Blog" Width="72px" />
```

11. Add a couple of breaks for spacing purposes.

```
<br /><br />
```

12. Add a results label that displays the blog information. It will also display a message if there is an error.

```
<asp:Label ID="lblResult" runat="server" Width="604px" Height="113px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
Protected Sub btnGetBlog_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGetBlog.Click
```

2. Create the following objects.

```
Dim iPostNum As Integer = -1
Dim blogDataItem As New Ektron.Cms.BlogData
Dim blogApi As New Ektron.Cms.API.Content.Blog
Dim i As Integer
```

3. Check to make sure txtBlogId is numeric.

```
If Not IsNumeric(txtBlogId.Text) Then
    lblResult.Text = "Invalid Blog ID"
    Exit Sub
End If
```

4. Check to make sure txtDate is a date.

```
If Not IsDate(txtDate.Text) Then
    lblResult.Text = "Invalid Date"
    Exit Sub
End If
```

5. Create a Try/Catch.

6. Add a check to see which radio button is active and set the number of posts.

```
Try
   If Me.rb_forday.Checked = True Then
       iPostNum = -1
   ElseIf Me.rb_month.Checked = True Then
       iPostNum = -2
   ElseIf Me.rb_dayprior.Checked = True Then
       iPostNum = 0
   ElseIf Me.rb_Custom.Checked = True Then
       If IsNumeric(Me.txt_num.Text) AndAlso CInt(Me.txt_num.Text) > 0 Then
```

```
                iPostNum = Convert.ToInt32(Me.txt_num.Text)
            Else
                Me.txt_num.Text = "1"
                iPostNum = 1
            End If
        End If
```

7. Get the blog.

```
        blogDataItem = blogApi.GetBlog(CInt(txtBlogId.Text), txtDate.Text, iPostNum, 1033)
```

Note: To get the blog using the path instead of the ID use the following code. `blogDataItem = blogApi.GetBlog("~SiteRoot~\TestBlog", txtDate.Text, -1, 1033)`

8. Check to see if the blog exists.

```
        If (blogDataItem.Id = 0) Then
            lblResult.Text = "Error getting blog"
            Exit Sub
        End If
```

9. Display the results for the user.

```
        lblResult.Text = "Title: " & blogDataItem.Title & "<br/>"
        lblResult.Text &= "Tagline: " & blogDataItem.Tagline & "<br/>"
        If blogDataItem.Categories.Length <> 0 Then
            lblResult.Text &= "Categories: "
            For i = 0 To blogDataItem.Categories.Length - 1 Step 1
                lblResult.Text &= blogDataItem.Categories(i) & " "
            Next
            lblResult.Text &= "<br>"
        End If

        If blogDataItem.BlogRoll.Items.Length <> 0 Then
            lblResult.Text &= "Blog roll links: "
            For i = 0 To blogDataItem.BlogRoll.Items.Length - 1 Step 1
                lblResult.Text &= "<a href=""" & blogDataItem.BlogRoll.Items(i).URL & """
target=""mainFrame"">" & blogDataItem.BlogRoll.Items(i).LinkName & "</a>" & "  "
            Next
            lblResult.Text &= "<br>"
        End If
        lblResult.Text &= "Number of posts: " & blogDataItem.Content.Length & "<br/>"
```

10. If there is an error, let the user know what happened.

```
    Catch ex As Exception
            lblResult.Text = "Error getting blog"
    End Try
```

Blog.GetBlog Method (String, String, Integer, Integer)

Returns a BlogData class for a given path, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

C#

```
public BlogData GetBlog(String path, String StartDate, int MaxEntries, int LanguageID);
```

Visual Basic

```
Public Function GetBlog(ByVal path As String, ByVal StartDate As String, ByVal MaxEntries As
Integer, ByVal LanguageID As Integer) As BlogData
```

Parameters

Parameters	Description
ByVal path As String	The path of the blog. This is a string value that corresponds to the folder path of the blog.
ByVal StartDate As String	The date to start getting entries. Used if you want to get historical information. Gets information only for the date specified.

ByVal MaxEntries As Integer	The total number of posts to return. This is an integer value. If set to 0, the `postsvisible` property of the blog is used. If set to -1, the content is not returned, only the post info for the content.
ByVal LanguageID As Integer	The language ID for the content. An integer value.

Returns

For a given path, this returns either null or a BlogData class with the associated posts. This also includes the roll, categories, and post information.

Remarks

Developers are advised to check for null prior to using.

GetBlogData Method

This is the overview for the GetBlogData method overload.

Description

Overloaded Method to obtain a BlogData Class.

Overload List

	Name	Description
	Blog.GetBlogData (FolderData) (see page 835)	Returns the corresponding blog (if applicable) for a complete folderData object. If there is code that already leverages our FolderData class, this can piggyback atop it. **Returns only if the folder is a blog.** This function is overloaded. See Also: • BlogObject(Long DateTime) • BlogObject(Long DateTime Integer Integer) • BlogObject(Long)
	Blog.GetBlogData (Long) (see page 835)	Overloaded Method to obtain a BlogData Class. Accepts the ID of the blog. Returns either a populated BlogData class or null (nothing). This function is overloaded. See Also: • BlogObject(Long DateTime) • BlogObject(Long DateTime Integer Integer) • BlogObject(FolderData)
	Blog.GetBlogData (Long, DateTime) (see page 836)	Overloaded Method to obtain a BlogData Class. Accepts the ID of the blog, as well as the StartDate (as DateTime) for the posts. Returns either a populated BlogData class or null (nothing). This function is overloaded. See Also: • BlogObject(Long) • BlogObject(Long DateTime Integer Integer) • BlogObject(FolderData)

	Blog.GetBlogData (Long, DateTime, Integer, Integer) (see page 836)	Overloaded Method to obtain a BlogData Class. Accepts the ID of the blog, the StartDate (as DateTime) for the posts, the number of posts visible and the language ID. Returns either a populated BlogData class or null (nothing). This function is overloaded. See Also: • BlogObject(Long) • BlogObject(Long DateTime) • BlogObject(FolderData)

Blog.GetBlogData Method (FolderData)

Returns the corresponding blog (if applicable) for a complete folderData object. If there is code that already leverages our FolderData class, this can piggyback atop it. **Returns only if the folder is a blog.**

This function is overloaded. See Also:

- BlogObject(Long DateTime)
- BlogObject(Long DateTime Integer Integer)
- BlogObject(Long)

C#

```
public BlogData GetBlogData(FolderData folder_data);
```

Visual Basic

```
Public Function GetBlogData(ByVal folder_data As FolderData) As BlogData
```

Parameters

Parameters	Description
ByVal folder_data As FolderData	A populated FolderData class. (see FolderData)

Returns

A populated BlogData class or nothing

Remarks

Blog.GetBlogData Method (Long)

Overloaded Method to obtain a BlogData Class. Accepts the ID of the blog. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

- BlogObject(Long DateTime)
- BlogObject(Long DateTime Integer Integer)
- BlogObject(FolderData)

C#

```
public BlogData GetBlogData(Long BlogID);
```

Visual Basic

```
Public Function GetBlogData(ByVal BlogID As Long) As BlogData
```

Parameters

Parameters	Description
ByVal BlogID As Long	Numeric ID of the blog. This corresponds to the ID of the blog folder.

Returns

Either a populated BlogData class or null (nothing).

Remarks

Blog.GetBlogData Method (Long, DateTime)

Overloaded Method to obtain a BlogData Class. Accepts the ID of the blog, as well as the StartDate (as DateTime) for the posts. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

- BlogObject(Long)
- BlogObject(Long DateTime Integer Integer)
- BlogObject(FolderData)

C#

```
public BlogData GetBlogData(Long BlogID, DateTime StartDate);
```

Visual Basic

```
Public Function GetBlogData(ByVal BlogID As Long, ByVal StartDate As DateTime) As BlogData
```

Parameters

Parameters	Description
ByVal BlogID As Long	Numeric ID of the blog. This corresponds to the ID of the blog folder.
ByVal StartDate As DateTime	The DateTime value from which to get posts. You can obtain historical posts with this property, or leave it blank to begin now.

Returns

Either a populated BlogData class or null (nothing).

Remarks

Blog.GetBlogData Method (Long, DateTime, Integer, Integer)

Overloaded Method to obtain a BlogData Class. Accepts the ID of the blog, the StartDate (as DateTime) for the posts, the number of posts visible and the language ID. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

- BlogObject(Long)
- BlogObject(Long DateTime)
- BlogObject(FolderData)

C#

```
public BlogData GetBlogData(Long BlogID, DateTime StartDate, int Postsvisible, int LanguageID);
```

Visual Basic

```
Public Function GetBlogData(ByVal BlogID As Long, ByVal StartDate As DateTime, ByVal
Postsvisible As Integer, ByVal LanguageID As Integer) As BlogData
```

Parameters

Parameters	Description
ByVal BlogID As Long	Numeric ID of the blog. This corresponds to the ID of the blog folder.
ByVal StartDate As DateTime	The DateTime value from which to get posts. You can obtain historical posts with this property, or leave it blank to begin now.
ByVal Postsvisible As Integer	Integer number of posts to return. Leave as 0 to use the blog setting, or -1 to return none.
ByVal LanguageID As Integer	Integer ID of the language.

Returns

Either a populated BlogData class or null (nothing).

Remarks

Blog.GetBlogDataForPost Method

Returns the blog information for a blog post. This contains information about categories, whether comments are enabled, etc. Accepts the post ID.

C#

```
public BlogData GetBlogDataForPost(Long post_id);
```

Visual Basic

```
Public Function GetBlogDataForPost(ByVal post_id As Long) As BlogData
```

Parameters

Parameters	Description
ByVal post_id As Long	Numeric value of the post ID

Returns

BlogData class

Blog.GetBlogPostData Method

Gets the Post data for a specific blog post. Accepts the ID of the post. Will return tags, trackback, pingback, and categories for the post.

C#

```
public BlogPostData GetBlogPostData(Long id);
```

Visual Basic

```
Public Function GetBlogPostData(ByVal id As Long) As BlogPostData
```

Parameters

Parameters	Description
ByVal id As Long	Numeric value of the post ID.

Returns

BlogPostData Class.

GetBlogRoll Method

This is the overview for the GetBlogRoll method overload.

Description

Returns a populated blog roll class. This blog roll is the list of links for the blog. See the BlogRoll class for the properties and usage of the blog roll.

This is an overloaded method.

Overload List

	Name	Description
	Blog.GetBlogRoll (Long) (see page 838)	Returns a populated blog roll class. Accepts a blog ID value. See the BlogRoll (see page 59) class for more information.
	Blog.GetBlogRoll (String) (see page 838)	Loads a blog roll for a given path. The path is a string of the blog path, ex: `\Media\Blogs\MyBlog`. This blog roll is the list of links for the blog. See the BlogRoll (see page 59) class for the properties and usage of the blog roll. This is an overloaded method. See Also: GetBlogRoll(Long).

Blog.GetBlogRoll Method (Long)

Returns a populated blog roll class. Accepts a blog ID value. See the BlogRoll (see page 59) class for more information.

C#

```
public BlogRoll GetBlogRoll(Long blog_id);
```

Visual Basic

```
Public Function GetBlogRoll(ByVal blog_id As Long) As BlogRoll
```

Parameters

Parameters	Description
ByVal blog_id As Long	Numeric ID of the blog. This corresponds to the folder ID of the blog.

Returns

A populated BlogRoll (see page 59) class or a null (nothing).

Remarks

Developers are advised to check for null prior to using this class.

Blog.GetBlogRoll Method (String)

Loads a blog roll for a given path. The path is a string of the blog path, ex: `\Media\Blogs\MyBlog`. This blog roll is the list of links for the blog. See the BlogRoll (see page 59) class for the properties and usage of the blog roll.

This is an overloaded method. See Also: GetBlogRoll(Long).

C#

```
public BlogRoll GetBlogRoll(String Path);
```

Visual Basic

```
Public Function GetBlogRoll(ByVal Path As String) As BlogRoll
```

Parameters

Parameters	Description
ByVal Path As String	The path of the blog. This is a string value that corresponds to the folder path of the blog.

Returns

This method returns a populated BlogRoll (see page 59) class or null (nothing).

Remarks

When using, developers are advised to check for null prior to using.

Blog.GetPost Method

Gets a blog post for a given ID. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData Class.

C#

```
public ContentData GetPost(Long post_id, ref BlogPostData PostInfo);
```

Visual Basic

```
Public Function GetPost(ByVal post_id As Long, ByRef PostInfo As BlogPostData) As ContentData
```

Parameters

Parameters	Description
ByVal post_id As Long	Numeric ID for the post.
ByRef PostInfo As BlogPostData	BlogPostData class representing the blog specific post info. This is also in the metadata.

Returns

ContentData class with a ByRef BlogPostData

Blog.GetPostbyID Method

Gets a blog post for a given ID. Includes trackback, pinback, comments, and other information. Returns BlogPostData Class.

C#

```
public BlogPostData GetPostbyID(Long post_id);
```

Visual Basic

```
Public Function GetPostbyID(ByVal post_id As Long) As BlogPostData
```

Parameters

Parameters	Description
ByVal post_id As Long	Numeric ID for the post.

Returns

BlogPostData class

Blog.GetUserBlog Method

gets the user blog id

C#

```
public Long GetUserBlog(Long UserId);
```

Visual Basic

```
Public Function GetUserBlog(ByVal UserId As Long) As Long
```

Parameters

Parameters	Description
ByVal UserId As Long	userid

Blog.UpdateBlogProperties Method

Updates the properties of a blog. Accepts standard blogdata object. Returns boolean.

C#

```
public Boolean UpdateBlogProperties(BlogData blog);
```

Visual Basic

```
Public Function UpdateBlogProperties(ByVal blog As BlogData) As Boolean
```

Parameters

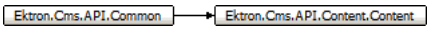
Parameters	Description
ByVal blog As BlogData	Information for the blog. Posts will be ignored, only blog properties will be updated.

Returns

Boolean value

Content Class

Class Hierarchy



C#

```
public class Content : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Content
Inherits Ektron.Cms.API.Common
```

File

Content.vb

Description

This is class Ektron.Cms.API.Content.Content.

Content Methods

The methods of the Content class are listed here.

Public Methods

	Name	Description
	AddAsset (see page 842)	Adds an asset to the CMS.
	AddContent (see page 842)	This is the overview for the AddContent method overload.
	AddXmlConfiguration (see page 847)	Adds an XML configuration for XML oriented content.
	CheckOutContent (see page 847)	This will check the content out to the current user if the content is currently checked in and the user has privileges to check it out.
	CopyContentToFolder (see page 847)	Copy the content into required folder.
	DeleteContentItem (see page 848)	Delete the content by the content's ID. *See Also:* The Example section.
	DeleteXmlConfiguration (see page 849)	Deletes an XML configuration.
	EditAsset (see page 849)	Edits an asset that was uploaded into the CMS.
	GetAddViewLanguage (see page 849)	Loads all the languages from the system for a given content ID and populates the Add/View drop down box with flag set on each item.

	GetAllComments (see page 850)	Loads all of the comments.
	GetAllTemplates (see page 850)	Loads all of the templates.
	GetAllUnassignedItemApprovals (see page 850)	Returns all of the unassigned user/groups for a given content ID.
	GetAllXmlConfigurations (see page 851)	This is the overview for the GetAllXmlConfigurations method overload.
	GetChildContent (see page 852)	This is the overview for the GetChildContent method overload.
	GetCommunityTemplate (see page 854)	Loads template for the given community type.
	GetContent (see page 854)	This is the overview for the GetContent method overload.
	GetContentByHistory (see page 857)	Retrieves the corresponding content by the history ID.
	GetContentDisplayLink (see page 858)	Function takes in the required information about the string then will decide the best link to use to display the content. It decides between Linkit, a generated quicklink or just the template file.
	GetContentForEditing (see page 858)	Loads the content details with checkout mode.
	GetContentReport (see page 858)	This is the overview for the GetContentReport method overload.
	GetContentState (see page 859)	Loads the content's details.
	GetContentStatus (see page 860)	This will return a single letter string which is the status of the piece of content. The letters returned are the same letters used in the Workarea view.
	GetDomain (see page 860)	Gets web domain for a piece of content.
	GetExpireContent (see page 860)	Loads all of the expired content.
	GetList (see page 860)	This is the overview for the GetList method overload.
	MoveContentToFolder (see page 861)	Moves content to a folder.
	PublishContent (see page 862)	Function takes in information about the content to be checked in and published. This goes through the normal approval chain. The content must be in a checked out state to the current logged in user for this to succeed.
	RelicateXmlConfiguration (see page 862)	**This method is obsolete. The recommended alternative is ReplicateXmlConfiguration (see page 863).** Replicate the XML configuration.
	RenameContent (see page 863)	Renames the content.
	ReplicateXmlConfiguration (see page 863)	Replicate the XML configuration.
	RestoreHistoryContent (see page 863)	Restores the content to a particular version of content.
	SaveContent (see page 863)	Updates the checked out content block. Only following properties can be updated: • contentData.Title • contentData.Teaser • contentData.Comment • contentData.GoLive • contentData.Html • contentData.EndDate • contentData.EndDateAction • contentData.ManualAlias
	SubmitForDelete (see page 864)	Delete the content by the content's Id and folder Id.
	UndoCheckout (see page 864)	This undoes the checkout for a piece of content which is checked out to the current user.
	UpdateContentMetaData (see page 865)	Updates a single piece of content metadata
	UpdatexmlCollectionPackage (see page 865)	Updates the XML collection package.

	UpdateXmlConfiguration (see page 865)	Updates the XML configuration details.

Content.AddAsset Method

Adds an asset to the CMS.

C#

```
public Long AddAsset(System.IO.Stream inputStream, AssetUpdateData assetdata);
```

Visual Basic

```
Public Function AddAsset(ByVal inputStream As System.IO.Stream, ByVal assetdata As
AssetUpdateData) As Long
```

Parameters

Parameters	Description
ByVal inputStream As System.IO.Stream	Input Stream pointing to the file to be uploaded
ByVal assetdata As AssetUpdateData	Asset (see page 822) details in AssetUpdateData (see page 1606)

Returns

Content (see page 840) Id

AddContent Method

This is the overview for the AddContent method overload.

Description

Adds Content to the CMS. This method is overloaded with three AddContent methods.

The first method allows you to add content and have the CMS assign it a content ID. The second, allows you to add content and assign it a content ID. The third, allows you to add content and assign it an XML ID and Template ID, while the CMS assigns it a content ID

Overload List

	Name	Description
	Content.AddContent (String, String, String, String, String, String, Long, Object, Object, Object) (see page 843)	Adds Content (see page 840) to the CMS. Upon successful completion, the CMS returns a Content (see page 840) ID. When content is added using this method, Web Alerts are **not** sent. *See Also:* The Example section.
	Content.AddContent (String, String, String, String, String, String, Long, Object, Object, Object, Long) (see page 845)	Adds Content (see page 840) to the CMS and allows you to assign it a Content (see page 840) ID. Once the content is added the new ID is returned. When content is added using this method, Web Alerts are **not** sent.
	Content.AddContent (String, String, String, String, String, String, Long, Object, Object, Object, Long, Long) (see page 845)	Adds Content (see page 840) to the CMS and allows you to assign the content an XML ID and Template ID. Once the content is added the new Content (see page 840) ID is returned. When content is added using this method, Web Alerts are **not** sent.

	Content.AddContent (String, String, String, String, String, String, Long, Object, Object, Object, Long, Long, Boolean) (see page 846)	Adds Content (see page 840) to the CMS and allows you to assign the content an XML ID and Template ID. Once the content is added the new Content (see page 840) ID is returned. When content is added using this method, Web Alerts are **not** sent.

Content.AddContent Method (String, String, String, String, String, String, Long, Object, Object, Object)

Adds Content (see page 840) to the CMS. Upon successful completion, the CMS returns a Content (see page 840) ID. When content is added using this method, Web Alerts are **not** sent.

See Also: The Example section.

C#

```
public Long AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, Long FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml);
```

Visual Basic

```
Public Function AddContent(ByVal ContentTitle As String, ByVal ContentComment As String, ByVal
ContentHtml As String, ByVal SearchText As String, ByVal SummaryHtml As String, ByVal
ContentLanguage As String, ByVal FolderId As Long, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object) As Long
```

Parameters

Parameters	Description
ByVal ContentTitle As String	Title of content to add.
ByVal ContentComment As String	Comments about the content being added.
ByVal ContentHtml As String	Content (see page 840) HTML to display.
ByVal SearchText As String	Search (see page 899) Text. This parameter allows for upgrades from CMS300. If you did not upgrade from CMS300, leave blank.
ByVal SummaryHtml As String	A summary of the HTML content.
ByVal ContentLanguage As String	The language to which the content is assigned.
ByVal FolderId As Long	Folder (see page 683) in which to place the content.
ByVal GoLive As Object	The date the content goes live on the site.
ByVal EndDate As Object	The date the content stops appearing on the site.
ByVal MetaInfoXml As Object	Meta Info XML

Returns

Long - ID of newly created content.

Example

The following example shows how to create a Web page from which a user can add content to the CMS. This example uses some standard drag and drop controls and a small section of VB code utilizing the AddContent method. This method uses the InternalAdmin to add the content. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a Label with the following information.
```
<asp:Label ID="Label1" runat="server" Text="Content Title: "></asp:Label>
```
2. Add a text box with the following information.
```
<asp:TextBox ID="txtTitle" runat="server"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Title Box.
```
<asp:RequiredFieldValidator ID="rfvTitle" runat="server" ControlToValidate="txtTitle"
   ErrorMessage="Please enter content title"></asp:RequiredFieldValidator>
```

4. Add a couple of breaks for spacing purposes.
```
<br /><br />
```
5. Add another label for the content text box.
```
<asp:Label ID="Label2" runat="server" Text="Content Text: " Width="87px"></asp:Label>
```
6. Add the text box where the content is entered. This is a MultiLine text box.
```
<asp:TextBox ID="txtContent" runat="server" Height="105px" TextMode="MultiLine"
Width="336px"></asp:TextBox>
```
7. Add a some more breaks for spacing purposes.
```
<br /><br />
```
8. Add a label for to show where to enter the Folder ID.
```
<asp:Label ID="Label3" runat="server" Text="Folder Id: " Width="65px"></asp:Label>
```
9. Add one last text box for the user to enter the Folder ID where the content will be stored.
   By default the text box is set to 0 (zero) which is the root folder.
```
<asp:TextBox ID="txtFolderId" runat="server" Width="44px">0</asp:TextBox>
```
10. Add another RequiredFieldValidator control. This validates that information is entered into the text box.
    Note: This does not validate that the folder actually exists in the CMS.
```
<asp:RequiredFieldValidator ID="rfvFolderId" runat="server" ControlToValidate="txtFolderId"
   ErrorMessage="Please enter the Folder Id"></asp:RequiredFieldValidator>
```
11. Add a break.
```
<br />
```
12. Add a label to describe the type of information that needs to be add to the folder ID box.
```
<asp:Label ID="Label4" runat="server" Font-Italic="True" Text="Enter a valid Folder ID from
the workarea, 0 points to the root folder."
   Width="483px"></asp:Label>
```
13. Add two more breaks.
```
<br /><br />
```
14. Add a button. We will set the button click event in the code behind.
```
<asp:Button ID="btnAdd" runat="server" Text="Add" />
```
15. Add two last breaks.
```
<br /><br />
```
16. Add a results label that lets a user know if the add succeeded.
```
<asp:Label ID="lblResult" runat="server"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.
```
Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
btnAdd.Click
```
2. Create an object for the content API (see page 669).
```
Dim contentApi As New Ektron.Cms.API.Content.Content
```
3. Check to make sure the folder ID is numeric.
```
If Not IsNumeric(txtFolderId.Text) Then
    lblResult.Text = "Invalid folder ID"
    Exit Sub
End If
```
4. Create an starting content ID of 0 (zero). When the content is added, it will be assigned an ID.
```
Dim ID As Integer = 0
```
5. Create a Try/Catch around the AddContent method. This allows you to catch any exceptions and helps when debugging.
```
Try
   ID = contentApi.AddContent(txtTitle.Text, "Content added by Example.aspx",
txtContent.Text, "", _
"", "1033", CInt(txtFolderId.Text), "", "", Nothing)
```
6. Add an If statement that lets the user know whether the content was added and the ID of the content.
```
   If (ID <> 0) Then
       lblResult.Text = "The above content was added into CMS400.NET with
ContentID = " & ID
   End If
```

7. If there is an error, let the user know what happened.

```
        Catch ex As Exception
             lblResult.Text = ex.Message
        End Try

End Sub
```

8. Add a page load event so that lblResult.Text is blank when the page first loads.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
   lblResult.Text = ""
End Sub
```

Content.AddContent Method (String, String, String, String, String, String, Long, Object, Object, Object, Long)

Adds Content (see page 840) to the CMS and allows you to assign it a Content (see page 840) ID. Once the content is added the new ID is returned. When content is added using this method, Web Alerts are **not** sent.

C#

```
public Long AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, Long FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml, Long ContentID);
```

Visual Basic

```
Public Function AddContent(ByVal ContentTitle As String, ByVal ContentComment As String, ByVal
ContentHtml As String, ByVal SearchText As String, ByVal SummaryHtml As String, ByVal
ContentLanguage As String, ByVal FolderId As Long, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object, ByVal ContentID As Long) As Long
```

Parameters

Parameters	Description
ByVal ContentTitle As String	Title of content to add.
ByVal ContentComment As String	Comments about the content being added.
ByVal ContentHtml As String	Content (see page 840) HTML to display.
ByVal SearchText As String	Search (see page 899) Text. This parameter allows for upgrades from CMS300. If you did not upgrade from CMS300, leave blank.
ByVal SummaryHtml As String	A summary of the HTML content.
ByVal ContentLanguage As String	The language to which the content is assigned.
ByVal FolderId As Long	Folder (see page 683) in which to place the content.
ByVal GoLive As Object	The date the content goes live on the site.
ByVal EndDate As Object	The date the content stops appearing on the site.
ByVal MetaInfoXml As Object	Meta Info XML
ByVal ContentID As Long	The ID you would like to assign the content.

Returns

Long - ID of newly created content.

Content.AddContent Method (String, String, String, String, String, String, Long, Object, Object, Object, Long, Long)

Adds Content (see page 840) to the CMS and allows you to assign the content an XML ID and Template ID. Once the content is added the new Content (see page 840) ID is returned. When content is added using this method, Web Alerts are **not** sent.

C#

```
public Long AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, Long FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml, Long XmlID, Long TemplateID);
```

Visual Basic

```
Public Function AddContent(ByVal ContentTitle As String, ByVal ContentComment As String, ByVal
```

```
ContentHtml As String, ByVal SearchText As String, ByVal SummaryHtml As String, ByVal
ContentLanguage As String, ByVal FolderId As Long, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object, ByVal XmlID As Long, ByVal TemplateID As Long) As Long
```

Parameters

Parameters	Description
ByVal ContentTitle As String	Title of content to add.
ByVal ContentComment As String	Comments about the content being added.
ByVal ContentHtml As String	Content (see page 840) HTML to display.
ByVal SearchText As String	Search (see page 899) Text. This parameter allows for upgrades from CMS300. If you did not upgrade from CMS300, leave blank.
ByVal SummaryHtml As String	A summary of the HTML content.
ByVal ContentLanguage As String	The language to which the content is assigned.
ByVal FolderId As Long	Folder (see page 683) in which to place the content.
ByVal GoLive As Object	The date the content goes live on the site.
ByVal EndDate As Object	The date the content stops appearing on the site.
ByVal MetaInfoXml As Object	Meta Info XML
ByVal XmlID As Long	XmlID
ByVal TemplateID As Long	Template ID

Returns

Long - ID of newly created content.

Content.AddContent Method (String, String, String, String, String, String, Long, Object, Object, Object, Long, Long, Boolean)

Adds Content (see page 840) to the CMS and allows you to assign the content an XML ID and Template ID. Once the content is added the new Content (see page 840) ID is returned. When content is added using this method, Web Alerts are **not** sent.

C#

```
public Long AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, Long FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml, Long XmlID, Long TemplateID, Boolean CleanHtml);
```

Visual Basic

```
Public Function AddContent(ByVal ContentTitle As String, ByVal ContentComment As String, ByVal
ContentHtml As String, ByVal SearchText As String, ByVal SummaryHtml As String, ByVal
ContentLanguage As String, ByVal FolderId As Long, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object, ByVal XmlID As Long, ByVal TemplateID As Long, ByVal
CleanHtml As Boolean) As Long
```

Parameters

Parameters	Description
ByVal ContentTitle As String	Title of content to add.
ByVal ContentComment As String	Comments about the content being added.
ByVal ContentHtml As String	Content (see page 840) HTML to display.
ByVal SearchText As String	Search (see page 899) Text. This parameter allows for upgrades from CMS300. If you did not upgrade from CMS300, leave blank.
ByVal SummaryHtml As String	A summary of the HTML content.
ByVal ContentLanguage As String	The language to which the content is assigned.
ByVal FolderId As Long	Folder (see page 683) in which to place the content.
ByVal GoLive As Object	The date the content goes live on the site.
ByVal EndDate As Object	The date the content stops appearing on the site.
ByVal MetaInfoXml As Object	Meta Info XML
ByVal XmlID As Long	XmlID
ByVal TemplateID As Long	Template ID
ByVal CleanHtml As Boolean	Clean ContentHTML and SummaryHTML with tidy.

Returns

Long - ID of newly created content.

Content.AddXmlConfiguration Method

Adds an XML configuration for XML oriented content.

C#

```
public Long AddXmlConfiguration(Collection pagedata);
```

Visual Basic

```
Public Function AddXmlConfiguration(ByVal pagedata As Collection) As Long
```

Parameters

Parameters	Description
ByVal pagedata As Collection	XML configuration data

Returns

A numeric value (Long).

Content.CheckOutContent Method

This will check the content out to the current user if the content is currently checked in and the user has privileges to check it out.

C#

```
public Boolean CheckOutContent(Long ContentId);
```

Visual Basic

```
Public Function CheckOutContent(ByVal ContentId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal ContentId As Long	Id of the content to check out

Returns

A Boolean representing the success of the check out

Content.CopyContentToFolder Method

Copy the content into required folder.

C#

```
public ContentData CopyContentToFolder(String Id, Long FID, String Lang, Boolean Publish);
```

Visual Basic

```
Public Function CopyContentToFolder(ByVal Id As String, ByVal FID As Long, ByVal Lang As
String, ByVal Publish As Boolean) As ContentData
```

Parameters

Parameters	Description
ByVal Id As String	Content (see page 840) ID
ByVal FID As Long	Folder (see page 683) ID
ByVal Lang As String	

Returns

Ektron.Cms.ContentData

Content.DeleteContentItem Method

Delete the content by the content's ID.

See Also: The Example section.

C#

```
public Boolean DeleteContentItem(Long Id);
```

Visual Basic

```
Public Function DeleteContentItem(ByVal Id As Long) As Boolean
```

Parameters

Parameters	Description
ByVal Id As Long	The content's ID.

Returns

Boolean - returns false if the delete succeeds.

Example

The following example shows how to create a Web page from which a user can delete content by entering its ID into a text box. This example uses some standard drag and drop controls and a small section of VB code utilizing the DeleteContentItem method. This method uses the InternalAdmin to delete the content. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a Label with the following information.
```
<asp:Label ID="Label1" runat="server" Text="Content ID: "></asp:Label>
```
2. Add a text box for the user to enter the content ID.
```
<asp:TextBox ID="txtContentId" runat="server" Width="44px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Content ID box.
```
<asp:RequiredFieldValidator ID="rfvContentId" runat="server"
ControlToValidate="txtContentId"
      ErrorMessage="Please enter content ID"></asp:RequiredFieldValidator>
```
4. Add a couple of breaks for spacing purposes.
```
<br /><br />
```
5. Add a button to delete the content. We will set the button click event in the code behind.
```
<asp:Button ID="btnDelete" runat="server" Text="Delete" />
```
6. Add a couple of more breaks for spacing purposes.
```
<br /><br />
```
7. Add a results label that lets a user know if the add succeeded.
```
<asp:Label ID="lblResult" runat="server"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.
```
Protected Sub btnDelete_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
btnDelete.Click
```
2. Create an object for the content API (see page 669).
```
Dim contentApi As New Ektron.Cms.API.Content.Content
```
3. Check to make sure the content ID is numeric.
```
If Not IsNumeric(txtContentId.Text) Then
    lblResult.Text = "Invalid content ID"
    Exit Sub
End If
```
4. Create an item *deleted* and set it to true. Note that DeleteContentItem returns false if the delete succeeds.

```
Dim deleted As Boolean = True
```

5. Create a Try/Catch around the DeleteContentItem method. This allows you to catch any exceptions and helps when debugging.

```
Try
   deleted = contentApi.DeleteContentItem(CInt(txtContentId.Text))
```

6. Add an If statement that lets the user know whether the content was deleted.

```
   If Not deleted Then
       lblResult.Text = "The above content was deleted from CMS400.Net"
   End If
```

7. If there is an error, let the user know what happened.

```
Catch ex As Exception
    lblResult.Text = ex.Message
End Try

End Sub
```

Content.DeleteXmlConfiguration Method

Deletes an XML configuration.

C#

```
public DeleteXmlConfiguration(Long Id);
```

Visual Basic

```
Public Sub DeleteXmlConfiguration(ByVal Id As Long)
```

Parameters

Parameters	Description
ByVal Id As Long	XML configuration ID

Content.EditAsset Method

Edits an asset that was uploaded into the CMS.

C#

```
public Boolean EditAsset(System.IO.Stream inputStream, AssetUpdateData assetdata);
```

Visual Basic

```
Public Function EditAsset(ByVal inputStream As System.IO.Stream, ByVal assetdata As
AssetUpdateData) As Boolean
```

Parameters

Parameters	Description
ByVal inputStream As System.IO.Stream	Input Stream pointing to the file to be uploaded if file is being overwritten
ByVal assetdata As AssetUpdateData	Asset (see page 822) details in AssetUpdateData (see page 1606)

Returns

A Boolean indicating success, False for success

Content.GetAddViewLanguage Method

Loads all the languages from the system for a given content ID and populates the Add/View drop down box with flag set on each item.

C#

```
public LanguageData[] GetAddViewLanguage(Long Id);
```

Visual Basic

```
Public Function GetAddViewLanguage(ByVal Id As Long) As LanguageData()
```

Parameters

Parameters	Description
ByVal Id As Long	Content (see page 840) ID

Returns

Ektron.Cms.LanguageData()

Content.GetAllComments Method

Loads all of the comments.

C#

```
public CommentData[] GetAllComments(Long KeyId, Long CommentId, Long RefId, String RefType,
Long UserId, String OrderBy);
```

Visual Basic

```
Public Function GetAllComments(ByVal KeyId As Long, ByVal CommentId As Long, ByVal RefId As
Long, ByVal RefType As String, ByVal UserId As Long, ByVal OrderBy As String) As CommentData()
```

Parameters

Parameters	Description
ByVal KeyId As Long	key ID
ByVal CommentId As Long	Comment ID
ByVal RefId As Long	Reference ID, (content or forms)
ByVal RefType As String	Reference type, C = content or T = Task (see page 909)
ByVal UserId As Long	The user ID
ByVal OrderBy As String	The order of the comments

Returns

Ektron.Cms.CommentData()

Content.GetAllTemplates Method

Loads all of the templates.

C#

```
public TemplateData[] GetAllTemplates(String OrderBy);
```

Visual Basic

```
Public Function GetAllTemplates(ByVal OrderBy As String) As TemplateData()
```

Parameters

Parameters	Description
ByVal OrderBy As String	Order by clause

Returns

Ektron.Cms.TemplateData()

Content.GetAllUnassignedItemApprovals Method

Returns all of the unassigned user/groups for a given content ID.

C#

```
public ApprovalData[] GetAllUnassignedItemApprovals(Long Id, String ItemType);
```

Visual Basic

```
Public Function GetAllUnassignedItemApprovals(ByVal Id As Long, ByVal ItemType As String) As
ApprovalData()
```

Parameters

Parameters	Description
ByVal Id As Long	The content's ID
ByVal ItemType As String	The type of user

Returns

Ektron.Cms.ApprovalData()

GetAllXmlConfigurations Method

This is the overview for the GetAllXmlConfigurations method overload.

Overload List

	Name	Description
	Content.GetAllXmlConfigurations (String) (see page 851)	Reads all of the XML configurations.
	Content.GetAllXmlConfigurations (String, Nullable(Of EkEnumeration.XmlConfigType)) (see page 851)	Reads all of the XML configurations.

Content.GetAllXmlConfigurations Method (String)

Reads all of the XML configurations.

C#

```
public XmlConfigData[] GetAllXmlConfigurations(String OrderBy);
```

Visual Basic

```
Public Function GetAllXmlConfigurations(ByVal OrderBy As String) As XmlConfigData()
```

Parameters

Parameters	Description
ByVal OrderBy As String	Order by clause

Returns

Ektron.Cms.XmlConfigData()

Content.GetAllXmlConfigurations Method (String, Nullable(Of EkEnumeration.XmlConfigType))

Reads all of the XML configurations.

C#

```
public XmlConfigData[] GetAllXmlConfigurations(String OrderBy, Nullable(Of
EkEnumeration.XmlConfigType) Type);
```

Visual Basic

```
Public Function GetAllXmlConfigurations(ByVal OrderBy As String, ByVal Type As Nullable(Of
EkEnumeration.XmlConfigType)) As XmlConfigData()
```

Parameters

Parameters	Description
ByVal OrderBy As String	Order by clause

Returns

Ektron.Cms.XmlConfigData()

GetChildContent Method

This is the overview for the GetChildContent method overload.

Overload List

	Name	Description
	Content.GetChildContent (Long) (see page 852)	Loads all of the content for the given folder.
	Content.GetChildContent (Long, Boolean) (see page 852)	Loads all of the content for the given folder.
	Content.GetChildContent (Long, Boolean, String) (see page 853)	Loads all of the content for the given folder.
	Content.GetChildContent (Long, Boolean, String, PagingInfo) (see page 853)	Loads all of the content for the given folder.

Content.GetChildContent Method (Long)

Loads all of the content for the given folder.

C#

```
public ContentData[] GetChildContent(Long Id);
```

Visual Basic

```
Public Function GetChildContent(ByVal Id As Long) As ContentData()
```

Parameters

Parameters	Description
ByVal Id As Long	The folder's ID.

Returns

Ektron.Cms.ContentData()

Content.GetChildContent Method (Long, Boolean)

Loads all of the content for the given folder.

C#

```
public ContentData[] GetChildContent(Long Id, Boolean Recursive);
```

Visual Basic

```
Public Function GetChildContent(ByVal Id As Long, ByVal Recursive As Boolean) As ContentData()
```

Parameters

Parameters	Description
ByVal Id As Long	The folder's ID.
ByVal Recursive As Boolean	True = recursive look up. False = none. Note: This parameter is not functional. Please use Content.GetList (see page 861)() api for recursive support.

Returns

Ektron.Cms.ContentData()

Content.GetChildContent Method (Long, Boolean, String)

Loads all of the content for the given folder.

C#

```
public ContentData[] GetChildContent(Long Id, Boolean Recursive, String OrderBy);
```

Visual Basic

```
Public Function GetChildContent(ByVal Id As Long, ByVal Recursive As Boolean, ByVal OrderBy As
String) As ContentData()
```

Parameters

Parameters	Description
ByVal Id As Long	The folder's ID.
ByVal Recursive As Boolean	True = recursive look up. False = none. Note: This parameter is not functional. Please use Content.GetList (see page 861)() api for recursive support.
ByVal OrderBy As String	An order by clause. Choices are: • **id** - the content ID. • **title** - the title of the content item. • **status** - whether the content is checked-in, published, checked-out, etc. • **language** - the language ID of the content. • **datecreated** - the date the content was created. • **datemodified** - the date the content was last modified. • **userid** - the ID of the user who last worked with the content.

Returns

Ektron.Cms.ContentData()

Content.GetChildContent Method (Long, Boolean, String, PagingInfo)

Loads all of the content for the given folder.

C#

```
public ContentData[] GetChildContent(Long Id, Boolean Recursive, String OrderBy, PagingInfo
pagingInfo);
```

Visual Basic

```
Public Function GetChildContent(ByVal Id As Long, ByVal Recursive As Boolean, ByVal OrderBy As
String, ByVal pagingInfo As PagingInfo) As ContentData()
```

Parameters

Parameters	Description
ByVal Id As Long	The folder's ID.
ByVal Recursive As Boolean	True = recursive look up. False = none.
ByVal OrderBy As String	An order by clause. Choices are: • **id** - the content ID. • **title** - the title of the content item. • **status** - whether the content is checked-in, published, checked-out, etc. • **language** - the language ID of the content. • **datecreated** - the date the content was created. • **datemodified** - the date the content was last modified. • **userid** - the ID of the user who last worked with the content.
ByVal pagingInfo As PagingInfo	paging Information for the retrieval

Returns

Ektron.Cms.ContentData()

Content.GetCommunityTemplate Method

Loads template for the given community type.

C#

```
public TemplateData[] GetCommunityTemplate(Ektron.Cms.Common.EkEnumeration.TemplateType type);
```

Visual Basic

```
Public Function GetCommunityTemplate(ByVal type As
Ektron.Cms.Common.EkEnumeration.TemplateType) As TemplateData()
```

Parameters

Parameters	Description
ByVal type As Ektron.Cms.Common.EkEnumeration.TemplateType	type of template

Returns

Ektron.Cms.TemplateData()

GetContent Method

This is the overview for the GetContent method overload.

Description

Returns list of content. This function is overloaded.

Overload List

	Name	Description
	Content.GetContent (Ektron.Cms.Common.ContentRequest, System.Web.UI.Page) (see page 855)	Returns list of contents using content request object. Only returns content of type CMSContentType.Content (see page 840) and CMSContentType.Assets.
	Content.GetContent (Long) (see page 855)	Loads the content details using the content ID. Only returns content of type CMSContentType.Content (see page 840) and CMSContentType.Assets.

	Content.GetContent (Long, ContentAPI.ContentResultType) (see page 855)	Loads the content details using the content ID. Only returns content of type CMSContentType.Content (see page 840) and CMSContentType.Assets.
	Content.GetContent (Long, Ektron.Cms.Content.EkContent.ContentResultType) (see page 856)	Using the content's ID and the ContentType to get the content and return it as ContentData. Only returns content of type CMSContentType.Content (see page 840) and CMSContentType.Assets. *See Also:* The Example section.

Content.GetContent Method (Ektron.Cms.Common.ContentRequest, System.Web.UI.Page)

Returns list of contents using content request object. Only returns content of type CMSContentType.Content (see page 840) and CMSContentType.Assets.

C#

```
public Ektron.Cms.Common.ContentResult GetContent(Ektron.Cms.Common.ContentRequest req,
Optional ByVal System.Web.UI.Page page);
```

Visual Basic

```
Public Function GetContent(ByVal req As Ektron.Cms.Common.ContentRequest, Optional ByVal page
As System.Web.UI.Page = Nothing) As Ektron.Cms.Common.ContentResult
```

Parameters

Parameters	Description
ByVal req As Ektron.Cms.Common.ContentRequest	ContentRequest
Optional ByVal page As System.Web.UI.Page = Nothing	Page reference

Returns

ContentResult

Content.GetContent Method (Long)

Loads the content details using the content ID. Only returns content of type CMSContentType.Content (see page 840) and CMSContentType.Assets.

C#

```
public ContentData GetContent(Long Id);
```

Visual Basic

```
Public Function GetContent(ByVal Id As Long) As ContentData
```

Parameters

Parameters	Description
ByVal Id As Long	The content ID

Returns

Ektron.Cms.ContentData

Content.GetContent Method (Long, ContentAPI.ContentResultType)

Loads the content details using the content ID. Only returns content of type CMSContentType.Content (see page 840) and CMSContentType.Assets.

C#

```
public ContentData GetContent(Long Id, ContentAPI.ContentResultType Type);
```

Visual Basic

```
Public Function GetContent(ByVal Id As Long, ByVal Type As ContentAPI.ContentResultType) As
ContentData
```

Parameters

Parameters	Description
ByVal Id As Long	The content ID
ByVal Type As ContentAPI.ContentResultType	

Returns

Ektron.Cms.ContentData

Content.GetContent Method (Long, Ektron.Cms.Content.EkContent.ContentResultType)

Using the content's ID and the ContentType to get the content and return it as ContentData. Only returns content of type CMSContentType.Content (see page 840) and CMSContentType.Assets.

See Also: The Example section.

C#

```
public ContentData GetContent(Long Id, Ektron.Cms.Content.EkContent.ContentResultType
ContentType);
```

Visual Basic

```
Public Function GetContent(ByVal Id As Long, ByVal ContentType As
Ektron.Cms.Content.EkContent.ContentResultType) As ContentData
```

Parameters

Parameters	Description
ByVal Id As Long	The ID of the content to be returned.
ByVal ContentType As Ektron.Cms.Content.EkContent.ContentResultType	Whether the content is published or staged. Enter either: Ektron.Cms.Content.EkContent.ContentResultType.Published or Ektron.Cms.Content.EkContent.ContentResultType.Staged

Returns

ContentData

Example

The following example shows how to create a Web page from which a user can view a content block and its title by entering the content ID into a text box. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetContent method. This method uses the InternalAdmin to get the content. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a Label with the following information.
```
<asp:Label ID="Label1" runat="server" Text="Content ID: "></asp:Label>
```
2. Add a text box for the user to enter the content ID.
```
<asp:TextBox ID="txtContentId" runat="server" Width="44px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Content ID box.
```
<asp:RequiredFieldValidator ID="rfvContentId" runat="server"
ControlToValidate="txtContentId"
    ErrorMessage="Please enter content ID"></asp:RequiredFieldValidator>
```
4. Add a couple of breaks for spacing purposes.

```
<br /><br />
```

5. Add a button to get the content. We will set the button click event in the code behind.
```
<asp:Button ID="btnGetContent" runat="server" Text="Get Content" />
```
6. Add a couple of more breaks for spacing purposes.
```
<br /><br />
```
7. Add a results label where the content will appear. This also serves as a place to display an error if necessary.
```
<asp:Label ID="lblResult" runat="server" Width="175px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.
```
   Protected Sub btnGetContent_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGetContent.Click
```
2. Create an object for the content API (see page 669).
```
       Dim contentApi As New Ektron.Cms.API.Content.Content
```
3. Check to make sure the content ID is numeric.
```
           If Not IsNumeric(txtContentId.Text) Then
               lblResult.Text = "Invalid content ID"
               Exit Sub
           End If
```
4. Create an object for to hold the content data that is returned.
```
       Dim contentItem As Ektron.Cms.ContentData
```
5. Create a Try/Catch around the GetContent method. This allows you to catch any exceptions and helps when debugging.
```
       Try
          contentItem = contentApi.GetContent(CInt(txtContentId.Text),
Ektron.Cms.Content.EkContent.ContentResultType.Published)
```
6. Add an If statement that displays the content or lets the user know if the content ID does not exist.
```
          If (contentItem Is Nothing) Then
             lblResult.Text = "ContentBlock does not exist"
          Else
             lblResult.Text = contentItem.Title & "<br/>" & contentItem.Html
          End If
```
7. If there is an error, let the user know what happened.
```
       Catch ex As Exception
           lblResult.Text = ex.Message
       End Try

   End Sub
```

Content.GetContentByHistory Method

Retrieves the corresponding content by the history ID.

C#

```
public ContentData GetContentByHistory(Long Id);
```

Visual Basic

```
Public Function GetContentByHistory(ByVal Id As Long) As ContentData
```

Parameters

Parameters	Description
ByVal Id As Long	The history ID

Returns

Ektron.Cms.ContentData

Content.GetContentDisplayLink Method

Function takes in the required information about the string then will decide the best link to use to display the content. It decides between Linkit, a generated quicklink or just the template file.

C#

```
public String GetContentDisplayLink(int ContentType, String ContentId, String ContentLanguage,
String FolderId, String host, String StagingDomain, String ProductionDomain);
```

Visual Basic

```
Public Function GetContentDisplayLink(ByVal ContentType As Integer, ByVal ContentId As String,
ByVal ContentLanguage As String, ByVal FolderId As String, ByVal host As String, ByVal
StagingDomain As String, ByVal ProductionDomain As String) As String
```

Parameters

Parameters	Description
ByVal ContentType As Integer	Integer describing the content's type (word doc, multimedia, etc.)
ByVal ContentId As String	A string value of the content's ID number
ByVal ContentLanguage As String	A string value of the content's language number
ByVal FolderId As String	A string value of the content's folder ID number
ByVal host As String	A string value of the host name to build the link out of
ByVal StagingDomain As String	A string value of the content's staging domain
ByVal ProductionDomain As String	A string value of the content's production domain

Returns

The page link as a String

Content.GetContentForEditing Method

Loads the content details with checkout mode.

C#

```
public ContentEditData GetContentForEditing(Long Id);
```

Visual Basic

```
Public Function GetContentForEditing(ByVal Id As Long) As ContentEditData
```

Parameters

Parameters	Description
ByVal Id As Long	The content ID.

Returns

Ektron.Cms.ContentEditData

GetContentReport Method

This is the overview for the GetContentReport method overload.

Overload List

	Name	Description
	Content.GetContentReport (Collection) (see page 859)	Loads the content data for the given constraints.

	Content.GetContentReport (ContentReportData) (see page 859)	Loads the content data for the given constraints.

Content.GetContentReport Method (Collection)

Loads the content data for the given constraints.

C#

```
public ContentData[] GetContentReport(Collection pagedata);
```

Visual Basic

```
Public Function GetContentReport(ByVal pagedata As Collection) As ContentData()
```

Parameters

Parameters	Description
ByVal pagedata As Collection	required report data

Returns

Ektron.Cms.ContentData()

Content.GetContentReport Method (ContentReportData)

Loads the content data for the given constraints.

C#

```
public ContentData[] GetContentReport(ContentReportData ContReportData);
```

Visual Basic

```
Public Function GetContentReport(ByVal ContReportData As ContentReportData) As ContentData()
```

Parameters

Parameters	Description
ByVal ContReportData As ContentReportData	required report data

Returns

Ektron.Cms.ContentData()

Content.GetContentState Method

Loads the content's details.

C#

```
public ContentStateData GetContentState(Long Id);
```

Visual Basic

```
Public Function GetContentState(ByVal Id As Long) As ContentStateData
```

Parameters

Parameters	Description
ByVal Id As Long	The content's ID

Returns

Ektron.Cms.ContentStateData

Content.GetContentStatus Method

This will return a single letter string which is the status of the piece of content. The letters returned are the same letters used in the Workarea view.

C#

```
public String GetContentStatus(Long ContentId);
```

Visual Basic

```
Public Function GetContentStatus(ByVal ContentId As Long) As String
```

Parameters

Parameters	Description
ByVal ContentId As Long	Id of the content to return the status of

Returns

A string with the single letter status of the content

Content.GetDomain Method

Gets web domain for a piece of content.

C#

```
public String GetDomain(Long ContentId);
```

Visual Basic

```
Public Function GetDomain(ByVal ContentId As Long) As String
```

Parameters

Parameters	Description
ByVal ContentId As Long	The content's ID

Returns

String

Content.GetExpireContent Method

Loads all of the expired content.

C#

```
public ContentData[] GetExpireContent(Collection pagedata);
```

Visual Basic

```
Public Function GetExpireContent(ByVal pagedata As Collection) As ContentData()
```

Parameters

Parameters	Description
ByVal pagedata As Collection	Required data for the report as a value, key pair

Returns

Ektron.Cms.ContentData()

GetList Method

This is the overview for the GetList method overload.

Overload List

	Name	Description
	Content.GetList (Criteria(Of ContentProperty)) (see page 861)	Returns a list of content based upon the supplied criteria.
	Content.GetList (Criteria(Of ContentProperty), Boolean) (see page 861)	Returns a list of content based upon the supplied criteria.

Content.GetList Method (Criteria(Of ContentProperty))

Returns a list of content based upon the supplied criteria.

C#

```
public List(Of ContentData) GetList(Criteria(Of ContentProperty) criteria);
```

Visual Basic

```
Public Function GetList(ByVal criteria As Criteria(Of ContentProperty)) As List(Of ContentData)
```

Parameters

Parameters	Description
ByVal criteria As Criteria(Of ContentProperty)	criteria used to restrict content retrieval.

Content.GetList Method (Criteria(Of ContentProperty), Boolean)

Returns a list of content based upon the supplied criteria.

C#

```
public List(Of ContentData) GetList(Criteria(Of ContentProperty) criteria, Boolean
folderRecursive);
```

Visual Basic

```
Public Function GetList(ByVal criteria As Criteria(Of ContentProperty), ByVal folderRecursive
As Boolean) As List(Of ContentData)
```

Parameters

Parameters	Description
ByVal criteria As Criteria(Of ContentProperty)	criteria used to restrict content retrieval.
ByVal folderRecursive As Boolean	If true, retrieves all recursive folders based upon folderId in criteria filter. This parameter is only relative if the criteria includes a FolderId filter.

Content.MoveContentToFolder Method

Moves content to a folder.

C#

```
public ContentData MoveContentToFolder(String cId, String cLanguages, Long FID);
```

Visual Basic

```
Public Function MoveContentToFolder(ByVal cId As String, ByVal cLanguages As String, ByVal FID
As Long) As ContentData
```

Parameters

Parameters	Description
ByVal cId As String	The content's ID
ByVal cLanguages As String	The content's language
ByVal FID As Long	The folder's ID

Returns

Ektron.Cms.ContentData

Content.PublishContent Method

Function takes in information about the content to be checked in and published. This goes through the normal approval chain. The content must be in a checked out state to the current logged in user for this to succeed.

C#

```
public Boolean PublishContent(Long ContentId, Long FolderId, int ContentLanguage, String
dontCreateTask, Long UserID, String TaskTitle);
```

Visual Basic

```
Public Function PublishContent(ByVal ContentId As Long, ByVal FolderId As Long, ByVal
ContentLanguage As Integer, ByVal dontCreateTask As String, ByVal UserID As Long, ByVal
TaskTitle As String) As Boolean
```

Parameters

Parameters	Description
ByVal ContentId As Long	Numeric value of the content's ID. (Long)
ByVal FolderId As Long	Numeric value of the content folder's ID. (Long)
ByVal ContentLanguage As Integer	Integer value of the content's language ID.
ByVal dontCreateTask As String	Set to **""** to create a task for this submit. If anything else appears, a task is not created.
ByVal UserID As Long	The ID of the user assigning the task (current user). This is needed only when the dontCreateTask property is set to "". Otherwise, this property is ingnored.
ByVal TaskTitle As String	Title for the task. This is needed only when the dontCreateTask property is set to "". Otherwise, this property is ingnored.

Returns

A Boolean indicating success or not.

Content.RelicateXmlConfiguration Method

This method is obsolete. The recommended alternative is ReplicateXmlConfiguration (see page 863).

Replicate the XML configuration.

C#

```
[Obsolete("The recommended alternative is ReplicateXmlConfiguration(ByVal Id As Long, ByVal
Title As String)")]
public Long RelicateXmlConfiguration(Long Id, String Title);
```

Visual Basic

```
<Obsolete("The recommended alternative is ReplicateXmlConfiguration(ByVal Id As Long, ByVal
Title As String)")>
Public Function RelicateXmlConfiguration(ByVal Id As Long, ByVal Title As String) As Long
```

Parameters

Parameters	Description
ByVal Id As Long	The inherited XML ID
ByVal Title As String	Title

Returns

Long

Content.RenameContent Method

Renames the content.

C#

```
public ContentData RenameContent(Long cId, String title);
```

Visual Basic

```
Public Function RenameContent(ByVal cId As Long, ByVal title As String) As ContentData
```

Parameters

Parameters	Description
ByVal cId As Long	The content's ID
ByVal title As String	The content's new title

Returns

Boolean

Content.ReplicateXmlConfiguration Method

Replicate the XML configuration.

C#

```
public Long ReplicateXmlConfiguration(Long Id, String Title);
```

Visual Basic

```
Public Function ReplicateXmlConfiguration(ByVal Id As Long, ByVal Title As String) As Long
```

Parameters

Parameters	Description
ByVal Id As Long	The inherited XML ID.
ByVal Title As String	The title of the XML configuration.

Returns

A numeric value. (Long)

Content.RestoreHistoryContent Method

Restores the content to a particular version of content.

C#

```
public RestoreHistoryContent(Long Id);
```

Visual Basic

```
Public Sub RestoreHistoryContent(ByVal Id As Long)
```

Parameters

Parameters	Description
ByVal Id As Long	The content's history ID

Content.SaveContent Method

Updates the checked out content block. Only following properties can be updated:

- contentData.Title
- contentData.Teaser
- contentData.Comment
- contentData.GoLive
- contentData.Html
- contentData.EndDate
- contentData.EndDateAction
- contentData.ManualAlias

C#

```
public Boolean SaveContent(ContentEditData contentEditData);
```

Visual Basic

```
Public Function SaveContent(ByVal contentEditData As ContentEditData) As Boolean
```

Parameters

Parameters	Description
ByVal contentEditData As ContentEditData	_nt_

Returns

False if successful

Remarks

Content (see page 840) must be in the checked out state. If you are updating an existing asset you mus make sure to set the FileChanged property to False in the contentEditData

Content.SubmitForDelete Method

Delete the content by the content's Id and folder Id.

C#

```
public Boolean SubmitForDelete(Long cId, Long fId);
```

Visual Basic

```
Public Function SubmitForDelete(ByVal cId As Long, ByVal fId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal cId As Long	The content's Id
ByVal fId As Long	The content's Folder (see page 683) Id

Returns

Boolean

Content.UndoCheckout Method

This undoes the checkout for a piece of content which is checked out to the current user.

C#

```
public Boolean UndoCheckout(Long ContentId);
```

Visual Basic

```
Public Function UndoCheckout(ByVal ContentId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal ContentId As Long	Id of the content for which to undo check out

Returns

A Boolean representing the success of the undo.

Content.UpdateContentMetaData Method

Updates a single piece of content metadata

C#

```
public Boolean UpdateContentMetaData(Long ContentID, Long MetaID, String MetaText);
```

Visual Basic

```
Public Function UpdateContentMetaData(ByVal ContentID As Long, ByVal MetaID As Long, ByVal
MetaText As String) As Boolean
```

Parameters

Parameters	Description
ByVal ContentID As Long	The Content (see page 840) ID
ByVal MetaID As Long	The Metadata (see page 714) Type ID
ByVal MetaText As String	The new text to assign

Returns

boolean

Content.UpdatexmlCollectionPackage Method

Updates the XML collection package.

C#

```
public UpdatexmlCollectionPackage(Collection package);
```

Visual Basic

```
Public Sub UpdatexmlCollectionPackage(ByVal package As Collection)
```

Parameters

Parameters	Description
ByVal package As Collection	The XML details

Content.UpdateXmlConfiguration Method

Updates the XML configuration details.

C#

```
public UpdateXmlConfiguration(Collection pagedata);
```

Visual Basic

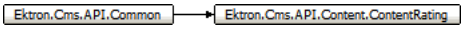
```
Public Sub UpdateXmlConfiguration(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
ByVal pagedata As Collection	The XML configuration details

ContentRating Class

Class Hierarchy



C#

```
public class ContentRating : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class ContentRating
Inherits Ektron.Cms.API.Common
```

File

ContentRating.vb

Description

This is class Ektron.Cms.API.Content.ContentRating.

ContentRating Methods

The methods of the ContentRating class are listed here.

Public Methods

	Name	Description
	GetContentRating (see page 866)	Constructs a string that represents a content rating element on a web site. This is composed of a rating dialog and a graph representing current ratings. Note that the content rating server control wraps this functionality.
	GetContentRatingAddRating (see page 868)	Adds a content rating to the database. Internally a check is done to ensure the user has not already rated the content, and passes down the rating and any comments the user supplied.
	GetContentRatingResults (see page 869)	Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content from the range of StartDate to EndDate. In order to get the full range of ratings, pass DateTime.Min and DateTime.Max as arguments.
	GetContentRatingStatistics (see page 869)	This is the overview for the GetContentRatingStatistics method overload.
	PurgeContentRatings (see page 870)	Purges all ratings of a particular content block over a defined date/time range.

ContentRating.GetContentRating Method

Constructs a string that represents a content rating element on a web site. This is composed of a rating dialog and a graph representing current ratings. Note that the content rating server control wraps this functionality.

C#

```
public String GetContentRating(Long contentID, System.Web.UI.Page currentPage, String key,
ContentRatingLabels Labels, ContentAPI.RatingGraphDisplayType initialGraph,
ContentAPI.RatingGraphDisplayType ratingCompleteGraph, int padding, System.Drawing.Color
bgColor, System.Drawing.Color barColor, System.Drawing.Color TextColor);
```

Visual Basic

```
Public Function GetContentRating(ByVal contentID As Long, ByVal currentPage As
System.Web.UI.Page, ByVal key As String, ByVal Labels As ContentRatingLabels, ByVal
initialGraph As ContentAPI.RatingGraphDisplayType, ByVal ratingCompleteGraph As
ContentAPI.RatingGraphDisplayType, ByVal padding As Integer, ByVal bgColor As
System.Drawing.Color, ByVal barColor As System.Drawing.Color, ByVal TextColor As
System.Drawing.Color) As String
```

Parameters

Parameters	Description
ByVal contentID As Long	The numeric ID associated with the content block being rated.

ByVal currentPage As System.Web.UI.Page	A reference to the System.Web.UI.Page object representing the active web page.
ByVal key As String	Specifies a unique ID of this element. No two content rating controls on the same page should specify the same key.
ByVal Labels As ContentRatingLabels	Specifies the labels of the control to be displayed. See Ektron.Cms.ContentRatingLabels (see page 1611)
ByVal initialGraph As ContentAPI.RatingGraphDisplayType	Specifies to show the initial (non-rated) graph to be horizontal to the control, below it, or not at all.
ByVal ratingCompleteGraph As ContentAPI.RatingGraphDisplayType	Specifies to show the complete (rated) graph to be horizontal to the control, below it, or not at all.
ByVal padding As Integer	Specifies the number of pixels between the rating control and the graph.
ByVal bgColor As System.Drawing.Color	Specifies the background color of the graph.
ByVal barColor As System.Drawing.Color	Specifies the bar color of the graph.
ByVal TextColor As System.Drawing.Color	Specifies the text color of the graph.

Returns

Returns a string representing the final control.

Remarks

Note that the ContentRating (see page 866) server control nicely wraps this function.

Example

The following example shows how to create a Web page that retrieves a content rating element used to rate content. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetContentRating method. This method uses the InternalAdmin to retrieve the content. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a Label with the following information.
```
<asp:Label ID="Label6" runat="server" Text="Content ID:" Width="76px"></asp:Label>
```
2. Add a text box for the user to enter the content ID.
```
<asp:TextBox ID="txtId" runat="server" Width="55px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Content ID box.
```
<asp:RequiredFieldValidator ID="rfvId" runat="server" ControlToValidate="txtId"
ErrorMessage="Please enter content id" Width="140px"></asp:RequiredFieldValidator>
```
4. Add a couple of breaks for spacing purposes.
```
<br /><br />
```
5. Add a button to submit the content ID and retrieve the content rating.
```
<asp:Button ID="btnGet" runat="server" Text="Get ContentRating" Width="115px" />
```
6. Add a couple of more breaks for spacing purposes.
```
<br /><br />
```
7. Add a results label to display the content rating.
```
<asp:Label ID="lblResult" runat="server" Width="502px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add the following Imports statement to the top of the page.
```
Imports Ektron.Cms
```
2. Add a page load event that makes sure lblResult.Text is blank.
```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
    lblResult.Text = ""
End Sub
```
3. Add a button click event.
```
Protected Sub btnGet_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGet.Click
```
4. Check to make sure the content ID is numeric.
```
    If Not IsNumeric(txtId.Text) Then
        lblResult.Text = "Invalid Content ID"
```

```
            Exit Sub
        End If
```

5. Create the following objects.

```
        Dim ratingApi As New Ektron.Cms.API.Content.ContentRating
        Dim labels As New Ektron.Cms.ContentRatingLabels
        Dim str As String
```

6. Define the labels for use in the content rating.

```
        labels.TotalRatingsLabel = "Total number of ratings"
        labels.RatingLevelLabel = "Rating Level: "
        labels.AlreadyRatedMessage = "You have rated this content"
        labels.GoodLabel = "Good"
        labels.BadLabel = "Bad"
        labels.FeedbackHeader = "How would you rate this content?"
        labels.UserCommentsHeader = "Enter any feedback below."
        labels.CharsRemainingLabel = "Remaining Characters: "
```

7. Create a Try/Catch around the GetContentRating method. This allows you to catch any exceptions and helps when debugging.

```
        Try
           str = ratingApi.GetContentRating(CInt(txtId.Text), Me.Page, "", labels,
ContentAPI.RatingGraphDisplayType.Horizontal, _
                 ContentAPI.RatingGraphDisplayType.Horizontal, 10, Drawing.Color.Aqua,
Drawing.Color.Red, Drawing.Color.Black)
```

8. Display the ratings.

```
        If str.Length > 0 Then
           lblResult.Text = str
        Else
           lblResult.Text = "No content rating information available"
        End If
```

9. If there is an error, let the user know what happened.

```
        Catch ex As Exception
            lblResult.Text = ex.Message
        End Try

    End Sub
```

ContentRating.GetContentRatingAddRating Method

Adds a content rating to the database. Internally a check is done to ensure the user has not already rated the content, and passes down the rating and any comments the user supplied.

C#

```
public GetContentRatingAddRating(Long contentid, Long userid, int userRating, String
userComments);
```

Visual Basic

```
Public Sub GetContentRatingAddRating(ByVal contentid As Long, ByVal userid As Long, ByVal
userRating As Integer, ByVal userComments As String)
```

Parameters

Parameters	Description
ByVal contentid As Long	The numeric ID associated with the content block being rated.
ByVal userid As Long	The unique global identifier stored as a cookie by the user.
ByVal userRating As Integer	The rating the user supplies (1-9).
ByVal userComments As String	Optional comments that the user provides.

ContentRating.GetContentRatingResults Method

Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content from the range of StartDate to EndDate. In order to get the full range of ratings, pass DateTime.Min and DateTime.Max as arguments.

C#

```
public System.Data.DataSet GetContentRatingResults(Long contentID, DateTime startDate,
DateTime endDate);
```

Visual Basic

```
Public Function GetContentRatingResults(ByVal contentID As Long, ByVal startDate As DateTime,
ByVal endDate As DateTime) As System.Data.DataSet
```

Parameters

Parameters	Description
ByVal contentID As Long	The numeric ID associated with the content block whose ratings are being queried.
ByVal startDate As DateTime	The start date of a range of dates to query over.
ByVal endDate As DateTime	The end date of a range of dates to query over.

Returns

(Username - String, RatingDate - DateTime, Rating - int, Comments - String)

GetContentRatingStatistics Method

This is the overview for the GetContentRatingStatistics method overload.

Description

Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate.

This method is overloaded.

Overload List

	Name	Description
	ContentRating.GetContentRatingStatistics (Long, DateTime, DateTime) (see page 869)	Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate. This function is overloaded. See Also: GetContentRatingStatistics(Long Long String)
	ContentRating.GetContentRatingStatistics (Long, Long, String) (see page 870)	Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate. This function is overloaded. See Also: GetContentRatingStatistics(Long DateTime DateTime)

ContentRating.GetContentRatingStatistics Method (Long, DateTime, DateTime)

Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate.

This function is overloaded. See Also: GetContentRatingStatistics(Long Long String)

C#

```
public Collection GetContentRatingStatistics(Long contentID, DateTime startDate, DateTime
```

```
endDate);
```

Visual Basic

```
Public Function GetContentRatingStatistics(ByVal contentID As Long, ByVal startDate As
DateTime, ByVal endDate As DateTime) As Collection
```

Parameters

Parameters	Description
ByVal contentID As Long	The numeric ID associated with the content block being rated.
ByVal startDate As DateTime	Bounds the result set by only ratings that occurred after this date.
ByVal endDate As DateTime	Bounds the result set by only ratings that occurred before this date.

Returns

Returns a collection of data with keys "total" => number of ratings, "sum" => sum of all ratings, "r1, "r2", "r3", "r4", "r5", "r6", "r7", "r8", "r9".

Remarks

Note that average rating = sum/total.

ContentRating.GetContentRatingStatistics Method (Long, Long, String)

Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate.

This function is overloaded. See Also: GetContentRatingStatistics(Long DateTime DateTime)

C#

```
public Collection GetContentRatingStatistics(Long contentid, Long user_id, String visitor_id);
```

Visual Basic

```
Public Function GetContentRatingStatistics(ByVal contentid As Long, ByVal user_id As Long,
ByVal visitor_id As String) As Collection
```

Parameters

Parameters	Description
ByVal contentid As Long	The numeric ID associated with the content block being rated.

Returns

Returns a collection of data with keys "total" => number of ratings, "sum" => sum of all ratings, "r1, "r2", "r3", "r4", "r5", "r6", "r7", "r8", "r9".

Remarks

Note that average rating = sum/total.

ContentRating.PurgeContentRatings Method

Purges all ratings of a particular content block over a defined date/time range.

C#

```
public PurgeContentRatings(Long contentid, DateTime startDate, DateTime endDate);
```

Visual Basic

```
Public Sub PurgeContentRatings(ByVal contentid As Long, ByVal startDate As DateTime, ByVal
endDate As DateTime)
```

Parameters

Parameters	Description
ByVal contentid As Long	The ID of the content that is having its ratings purged.
ByVal startDate As DateTime	The beginning of the date range of content to be purged.
ByVal endDate As DateTime	The end of the date range of content to be purged.

Form Class

Class Hierarchy

C#

```
public class Form : Content;
```

Visual Basic

```
Public Class Form
Inherits Content
```

File

Form.vb

Description

This is class Ektron.Cms.API.Content.Form.

Form Methods

The methods of the Form class are listed here.

Public Methods

	Name	Description
	AssignFormTask (see page 871)	Assigns a form submission task, as specified in FormInfo (see page 588), of the SubmittedDataXml.
	EmailFormFieldData (see page 872)	Sends an email, as specified in FormInfo (see page 588), of the SubmittedDataXml.
	GetAllForms (see page 872)	This is the overview for the GetAllForms method overload.
	GetForm (see page 873)	Retrieves information about the specified form.
	GetFormFieldData (see page 873)	This is the overview for the GetFormFieldData method overload.
	GetFormFieldList (see page 874)	Retrieves the list of form fields defined in the specified form.
	GetFormFieldQuestions (see page 875)	Retrieves the table of form fields and their questions in a specified form. In a CMS form, the form question is defined by the user in the descriptive name field in the form element dialog during design time.
	GetFormTitle (see page 875)	Same as GetFormById, but returns only the form's title.
	SubmitFormFieldData (see page 875)	Adds form data as if it was submitted during a browser post to the web server.
	UpdateFormFieldData (see page 876)	Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

Form.AssignFormTask Method

Assigns a form submission task, as specified in FormInfo (see page 588), of the SubmittedDataXml.

C#

```
public Boolean AssignFormTask(FormData FormInfo, String SubmittedDataXml);
```

Visual Basic

```
Public Function AssignFormTask(ByVal FormInfo As FormData, ByVal SubmittedDataXml As String)
```

```
As Boolean
```

Parameters

Parameters	Description
ByVal FormInfo As FormData	The properties of the form associated with the data. This information may be retrieved by calling GetFormById. Value Type: Ektron.Cms.FormData
ByVal SubmittedDataXml As String	The form data submitted in XML format. Value Type: System.String

Returns

Returns True if the task was successfully assigned. Value Type: System.Boolean

Remarks

This routine is called automatically when calling SubmitFormFieldData (see page 875) if SendNotification is True and the form is configured to assign a task when data is submitted. The SubmittedDataXml is typically retrieved by calling SubmitFormFieldData (see page 875).

Form.EmailFormFieldData Method

Sends an email, as specified in FormInfo (see page 588), of the SubmittedDataXml.

C#

```
public EmailFormFieldData(FormData FormInfo, String SubmittedDataXml);
```

Visual Basic

```
Public Sub EmailFormFieldData(ByVal FormInfo As FormData, ByVal SubmittedDataXml As String)
```

Parameters

Parameters	Description
ByVal FormInfo As FormData	The properties of the form associated with the data. This information may be retrieved by calling GetFormById. Value Type: Ektron.Cms.FormData
ByVal SubmittedDataXml As String	The form data submitted in XML format. Value Type: System.String

Remarks

This routine is called automatically when calling SubmitFormFieldData (see page 875) if SendNotification is True and the form is configured to send an email notification when data is submitted. The SubmittedDataXml is typically retrieved by calling SubmitFormFieldData (see page 875).

GetAllForms Method

This is the overview for the GetAllForms method overload.

Description

Retrieves an array of all the forms contained with the specified folder. This method is overloaded.

Overload List

	Name	Description
	Form.GetAllForms () (see page 873)	Returns all of the forms as an array of FormData.
	Form.GetAllForms (Long, String, Integer, Cms.ContentAPI.FormResultType) (see page 873)	Retrieves an array of all the forms contained with the specified folder. Forms may be limited to a single language and type.

Form.GetAllForms Method ()

Returns all of the forms as an array of FormData.

C#

```
public FormData[] GetAllForms();
```

Visual Basic

```
Public Function GetAllForms() As FormData()
```

Returns

Ektron.Cms.FormData()

Form.GetAllForms Method (Long, String, Integer, Cms.ContentAPI.FormResultType)

Retrieves an array of all the forms contained with the specified folder. Forms may be limited to a single language and type.

C#

```
public FormData[] GetAllForms(Long Id, String OrderBy, int LanguageId,
Cms.ContentAPI.FormResultType Type);
```

Visual Basic

```
Public Function GetAllForms(ByVal Id As Long, ByVal OrderBy As String, ByVal LanguageId As
Integer, ByVal Type As Cms.ContentAPI.FormResultType) As FormData()
```

Parameters

Parameters	Description
ByVal Id As Long	Specifies the folder.
ByVal OrderBy As String	Specifies how to sort the forms
ByVal LanguageId As Integer	Specifies a language or EkConstants.g_AllContentLanguages (-1) for all languages.
ByVal Type As Cms.ContentAPI.FormResultType	Specifies whether only active forms or archived forms should be returned.

Form.GetForm Method

Retrieves information about the specified form.

C#

```
public FormData GetForm(Long Id);
```

Visual Basic

```
Public Function GetForm(ByVal Id As Long) As FormData
```

Parameters

Parameters	Description
ByVal Id As Long	Specifies the form.

Returns

Returns information about the form.

GetFormFieldData Method

This is the overview for the GetFormFieldData method overload.

Description

Retrieves the submitted data for all fields from a specified form. This method is overloaded.

Overload List

	Name	Description
	Form.GetFormFieldData (Long, Long) (see page 874)	Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.
	Form.GetFormFieldData (Long, String, String) (see page 874)	Retrieves the submitted data for all fields from a specified form. All the data may be retrieved or just data submitted within a given date range.

Form.GetFormFieldData Method (Long, Long)

Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

C#

```
public FormSubmittedData GetFormFieldData(Long FormId, Long FormDataId);
```

Visual Basic

```
Public Function GetFormFieldData(ByVal FormId As Long, ByVal FormDataId As Long) As
FormSubmittedData
```

Parameters

Parameters	Description
ByVal FormId As Long	Specifies the form. Value Type: System.Int32
ByVal FormDataId As Long	Specifies the instance of submitted data. Value Type: System.Int32

Returns

Returns the submitted data or null/Nothing, if none found. Ektron.Cms.FormSubmittedData

Form.GetFormFieldData Method (Long, String, String)

Retrieves the submitted data for all fields from a specified form. All the data may be retrieved or just data submitted within a given date range.

C#

```
public FormSubmittedData[] GetFormFieldData(Long FormId, Optional ByVal String StartDate,
Optional ByVal String EndDate);
```

Visual Basic

```
Public Function GetFormFieldData(ByVal FormId As Long, Optional ByVal StartDate As String =
"", Optional ByVal EndDate As String = "") As FormSubmittedData()
```

Parameters

Parameters	Description
ByVal FormId As Long	Specifies the form. Value Type: System.Int32
Optional ByVal StartDate As String = ""	Optionally limit the data returned to include only those submitted on or after the given date. Value Type: System.String
Optional ByVal EndDate As String = ""	Optionally limit the data returned to include only those submitted on or before the given date. Value Type: System.String

Returns

Returns an array of submitted data. Ektron.Cms.FormSubmittedData

Form.GetFormFieldList Method

Retrieves the list of form fields defined in the specified form.

C#

```
public FormFieldList GetFormFieldList(Long FormId);
```

Visual Basic

```
Public Function GetFormFieldList(ByVal FormId As Long) As FormFieldList
```

Parameters

Parameters	Description
ByVal FormId As Long	Specifies the form. Value Type: System.Int32

Returns

The list for defined fields in the given form. Ektron.Cms.FormFieldList (see page 1636)

Form.GetFormFieldQuestions Method

Retrieves the table of form fields and their questions in a specified form. In a CMS form, the form question is defined by the user in the descriptive name field in the form element dialog during design time.

C#

```
public Hashtable GetFormFieldQuestions(Long FormId);
```

Visual Basic

```
Public Function GetFormFieldQuestions(ByVal FormId As Long) As Hashtable
```

Parameters

Parameters	Description
ByVal FormId As Long	Specifies the form. Value Type: System.Int32

Returns

A table of defined fields with the corresponding questions in a given form.

Remarks

These questions are mainly used for reporting.

Form.GetFormTitle Method

Same as GetFormById, but returns only the form's title.

C#

```
public String GetFormTitle(Long Id);
```

Visual Basic

```
Public Function GetFormTitle(ByVal Id As Long) As String
```

Parameters

Parameters	Description
ByVal Id As Long	Specifies the form.

Returns

The title of the specified form.

Form.SubmitFormFieldData Method

Adds form data as if it was submitted during a browser post to the web server.

C#

```
public SubmitFormFieldData(Long FormId, FormSubmittedData Data, Optional ByVal Boolean
SendNotification, [Out()] Optional ByRef String SubmittedDataXml);
```

Visual Basic

```
Public Sub SubmitFormFieldData(ByVal FormId As Long, ByVal Data As FormSubmittedData, Optional
ByVal SendNotification As Boolean = True, <Out()> Optional ByRef SubmittedDataXml As String =
"")
```

Parameters

Parameters	Description
ByVal FormId As Long	Specifies the form that corresponds to the data. Value Type: System.Int32
ByVal Data As FormSubmittedData	The data to submit. Value Type: Ektron.Cms.FormSubmittedData
Optional ByVal SendNotification As Boolean = True	(Optional) Specifies whether to send notifications that a form submission has occurred or not. For example, whether to send an email or assign a form submission task as configured for the form. Value Type: System.Boolean
Optional ByRef SubmittedDataXml As String = ""	(Optional) XML packet of submitted data. Reference Type: System.String

Form.UpdateFormFieldData Method

Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

C#

```
public UpdateFormFieldData(Long FormId, Long FormDataId, FormSubmittedData SubmittedData);
```

Visual Basic

```
Public Sub UpdateFormFieldData(ByVal FormId As Long, ByVal FormDataId As Long, ByVal
SubmittedData As FormSubmittedData)
```

Parameters

Parameters	Description
ByVal FormId As Long	Specifies the form. Value Type: System.Int32
ByVal FormDataId As Long	Specifies the instance of submitted data. Value Type: System.Int32
ByVal SubmittedData As FormSubmittedData	The field data values to be updated. Value Type: Ektron.Cms.FormSubmittedData

Taxonomy Class

Class Hierarchy

C#

```
public class Taxonomy : Content;
```

Visual Basic

```
Public Class Taxonomy
Inherits Content
```

File

Taxonomy.vb

Description

This is class Ektron.Cms.API.Content.Taxonomy.

Taxonomy Methods

The methods of the Taxonomy class are listed here.

Public Methods

	Name	Description
	AddTaxonomyItem (see page 877)	This is the overview for the AddTaxonomyItem method overload.
	AddTaxonomySyncFolder (see page 878)	Adds a folder entry into taxonomy.
	CloneTaxonomy (see page 878)	Copy or move a group of taxonomy categories from one taxonomy to another. For example, you want move categories one, two and three from taxonomy A to taxonomy B. Copying or moving a taxonomy category depends on whether the DeleteSource parameter is set True or False. If DeleteSouce is set to True, the taxonomy category is deleted. You can specify how many node levels are included by setting the Depth parameter. You can also choose whether to move/copy items in the category or just clone the category's structure.
	CreateTaxonomy (see page 879)	Creates a new taxonomy.
	DeleteTaxonomy (see page 879)	Deletes the existing taxonomy.
	GetTaxonomyIdByPath (see page 879)	Returns the taxonomy ID by a given path.
	ImportTaxonomy (see page 880)	This is the overview for the ImportTaxonomy method overload.
	LoadTaxonomy (see page 880)	Returns the TaxonomyData with all of its subcategory and its items.
	ReadAllAssignedCategory (see page 881)	Returns all the assigned categories for the content ID.
	ReadTaxonomy (see page 881)	Read the taxonomy data.
	RemoveTaxonomyFolder (see page 881)	Removes the sync up folder from taxonomy.
	RemoveTaxonomyItem (see page 882)	Removes an item from the taxonomy.
	ReOrderTaxonomyItems (see page 882)	Reorder the display of the taxonomy items.
	UpdateTaxonomy (see page 882)	Updates an existing taxonomy.

AddTaxonomyItem Method

This is the overview for the AddTaxonomyItem method overload.

Overload List

	Name	Description
	Taxonomy.AddTaxonomyItem (TaxonomyContentRequest) (see page 877)	Inserts an item into taxonomy.
	Taxonomy.AddTaxonomyItem (TaxonomyRequest) (see page 878)	Inserts an item into the taxonomy.

Taxonomy.AddTaxonomyItem Method (TaxonomyContentRequest)

Inserts an item into taxonomy.

C#

```
public AddTaxonomyItem(TaxonomyContentRequest request);
```

Visual Basic

```
Public Sub AddTaxonomyItem(ByVal request As TaxonomyContentRequest)
```

Parameters

Parameters	Description
ByVal request As TaxonomyContentRequest	TaxonomyContentRequest

Taxonomy.AddTaxonomyItem Method (TaxonomyRequest)

Inserts an item into the taxonomy.

C#

```
public AddTaxonomyItem(TaxonomyRequest request);
```

Visual Basic

```
Public Sub AddTaxonomyItem(ByVal request As TaxonomyRequest)
```

Parameters

Parameters	Description
ByVal request As TaxonomyRequest	TaxonomyRequest (see page 1643)

Taxonomy.AddTaxonomySyncFolder Method

Adds a folder entry into taxonomy.

C#

```
public AddTaxonomySyncFolder(TaxonomySyncRequest request);
```

Visual Basic

```
Public Sub AddTaxonomySyncFolder(ByVal request As TaxonomySyncRequest)
```

Parameters

Parameters	Description
ByVal request As TaxonomySyncRequest	TaxonomyRequest (see page 1643)

Taxonomy.CloneTaxonomy Method

Copy or move a group of taxonomy categories from one taxonomy to another. For example, you want move categories one, two and three from taxonomy A to taxonomy B. Copying or moving a taxonomy category depends on whether the DeleteSource parameter is set True or False. If DeleteSouce is set to True, the taxonomy category is deleted. You can specify how many node levels are included by setting the Depth parameter. You can also choose whether to move/copy items in the category or just clone the category's structure.

C#

```
public Long CloneTaxonomy(Long sourceId, Long destinationId, int LanguageId, int Depth,
Boolean WithItems, Boolean DeleteSource);
```

Visual Basic

```
Public Function CloneTaxonomy(ByVal sourceId As Long, ByVal destinationId As Long, ByVal
LanguageId As Integer, ByVal Depth As Integer, ByVal WithItems As Boolean, ByVal DeleteSource
As Boolean) As Long
```

Parameters

Parameters	Description
ByVal sourceId As Long	The ID of the source category being moved or copied.
ByVal destinationId As Long	The target ID where the category will be moved or copied.
ByVal LanguageId As Integer	The category's language ID.
ByVal Depth As Integer	Specify an integer that represents the amount of category sub levels to be copied or moved. For example, if you want to copy two category levels below the source category, enter **2**. To copy or move all sub categories enter **-1**.

ByVal WithItems As Boolean	Enter True to copy or move all items from the source structure. If you enter false, copy/move only the taxonomy category's structure.
ByVal DeleteSource As Boolean	Enter True to move the source category. This deletes the source category after the node is cloned. Set to false to copy the source category. This leaves the original category in its structure.

Returns

int

Taxonomy.CreateTaxonomy Method

Creates a new taxonomy.

C#

```
public Long CreateTaxonomy(TaxonomyData request);
```

Visual Basic

```
Public Function CreateTaxonomy(ByVal request As TaxonomyData) As Long
```

Parameters

Parameters	Description
ByVal request As TaxonomyData	TaxonomyData

Returns

Returns an ID for the newly created Taxonomy (see page 876).

Taxonomy.DeleteTaxonomy Method

Deletes the existing taxonomy.

C#

```
public DeleteTaxonomy(TaxonomyRequest request);
```

Visual Basic

```
Public Sub DeleteTaxonomy(ByVal request As TaxonomyRequest)
```

Parameters

Parameters	Description
ByVal request As TaxonomyRequest	TaxonomyRequest (see page 1643)

Taxonomy.GetTaxonomyIdByPath Method

Returns the taxonomy ID by a given path.

C#

```
public Long GetTaxonomyIdByPath(String path);
```

Visual Basic

```
Public Function GetTaxonomyIdByPath(ByVal path As String) As Long
```

Parameters

Parameters	Description
ByVal path As String	The path of the taxonomy category. For example, "\New Hampshire\Restaurants\Pizza"

Returns

Taxonomy (see page 876) ID

ImportTaxonomy Method

This is the overview for the ImportTaxonomy method overload.

Overload List

	Name	Description
	Taxonomy.ImportTaxonomy (String, String) (see page 880)	Creates a taxonomy using XML file.
	Taxonomy.ImportTaxonomy (String, String, List(Of TaxonomyType)) (see page 880)	Creates a taxonomy using an XML file.

Taxonomy.ImportTaxonomy Method (String, String)

Creates a taxonomy using XML file.

C#

```
public Long ImportTaxonomy(String TaxonomyXml, String Title);
```

Visual Basic

```
Public Function ImportTaxonomy(ByVal TaxonomyXml As String, ByVal Title As String) As Long
```

Parameters

Parameters	Description
ByVal TaxonomyXml As String	The XML being used to create a taxonomy.
ByVal Title As String	The title for the new taxonomy.

Returns

Returns a numeric ID for the newly created taxonomy.

Taxonomy.ImportTaxonomy Method (String, String, List(Of TaxonomyType))

Creates a taxonomy using an XML file.

C#

```
public Long ImportTaxonomy(String TaxonomyXml, String Title, List(Of TaxonomyType) config);
```

Visual Basic

```
Public Function ImportTaxonomy(ByVal TaxonomyXml As String, ByVal Title As String, ByVal
config As List(Of TaxonomyType)) As Long
```

Parameters

Parameters	Description
ByVal TaxonomyXml As String	The XML being used to create a taxonomy.
ByVal Title As String	The title for the new taxonomy.
ByVal config As List(Of TaxonomyType)	Configuration list that can be used.

Returns

Returns a numeric ID for the newly created taxonomy.

Taxonomy.LoadTaxonomy Method

Returns the TaxonomyData with all of its subcategory and its items.

C#

```
public TaxonomyData LoadTaxonomy(ref TaxonomyRequest request);
```

Visual Basic

```
Public Function LoadTaxonomy(ByRef request As TaxonomyRequest) As TaxonomyData
```

Parameters

Parameters	Description
ByRef request As TaxonomyRequest	TaxonomyRequest (see page 1643)

Returns

TaxonomyData

Taxonomy.ReadAllAssignedCategory Method

Returns all the assigned categories for the content ID.

C#

```
public TaxonomyBaseData[] ReadAllAssignedCategory(Long contentid);
```

Visual Basic

```
Public Function ReadAllAssignedCategory(ByVal contentid As Long) As TaxonomyBaseData()
```

Parameters

Parameters	Description
ByVal contentid As Long	A content's numeric ID.

Returns

An array of TaxonomyBaseData.

Taxonomy.ReadTaxonomy Method

Read the taxonomy data.

C#

```
public TaxonomyData ReadTaxonomy(ref TaxonomyRequest request);
```

Visual Basic

```
Public Function ReadTaxonomy(ByRef request As TaxonomyRequest) As TaxonomyData
```

Parameters

Parameters	Description
ByRef request As TaxonomyRequest	TaxonomyRequest (see page 1643)

Returns

Returns Taxonomydata

Taxonomy.RemoveTaxonomyFolder Method

Removes the sync up folder from taxonomy.

C#

```
public RemoveTaxonomyFolder(TaxonomySyncRequest request);
```

Visual Basic

```
Public Sub RemoveTaxonomyFolder(ByVal request As TaxonomySyncRequest)
```

Parameters

Parameters	Description
ByVal request As TaxonomySyncRequest	TaxonomyRequest (see page 1643)

Taxonomy.RemoveTaxonomyItem Method

Removes an item from the taxonomy.

C#

```
public RemoveTaxonomyItem(TaxonomyRequest request);
```

Visual Basic

```
Public Sub RemoveTaxonomyItem(ByVal request As TaxonomyRequest)
```

Parameters

Parameters	Description
ByVal request As TaxonomyRequest	TaxonomyRequest (see page 1643)

Taxonomy.ReOrderTaxonomyItems Method

Reorder the display of the taxonomy items.

C#

```
public ReOrderTaxonomyItems(TaxonomyRequest request);
```

Visual Basic

```
Public Sub ReOrderTaxonomyItems(ByVal request As TaxonomyRequest)
```

Parameters

Parameters	Description
ByVal request As TaxonomyRequest	TaxonomyReqeust

Taxonomy.UpdateTaxonomy Method

Updates an existing taxonomy.

C#

```
public UpdateTaxonomy(TaxonomyData request);
```

Visual Basic

```
Public Sub UpdateTaxonomy(ByVal request As TaxonomyData)
```

Parameters

Parameters	Description
ByVal request As TaxonomyData	TaxonomyData

ThreadedDiscussion Class

The ThreadedDiscussion class allows you to work with threaded discussion by adding and removing discussion boards, forums, and topics. It also allows you to subscribe or unsubscribe to a board, forum or topic. In addition, you can retrieve a specific topic or search a topic's replies.

The Threaded Discussion feature provides a discussion board where a forum can be created and topics can be discussed on your Web site. A site or membership user with Add Topic permission creates a topic and posts a starter question. Other site visitors can then reply to the topic.

Class Hierarchy

C#

```
public class ThreadedDiscussion : Content;
```

Visual Basic

```
Public Class ThreadedDiscussion
Inherits Content
```

File

ThreadedDiscussion.vb

ThreadedDiscussion Methods

The methods of the ThreadedDiscussion class are listed here.

Public Methods

	Name	Description
	AddBoard (see page 884)	Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns numeric ID of new discussion board.
	AddForum (see page 884)	Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns numeric ID of new discussion forum as Long. *See Also:* The Example section.
	AddReply (see page 886)	This is the overview for the AddReply method overload.
	AddTopic (see page 888)	This is the overview for the AddTopic method overload.
	DeleteBoard (see page 889)	Deletes a board with a given ID. This deletes the board as well as all forums, topics and posts under it. BoardID must be greater than 0.
	DeleteForum (see page 889)	Deletes a forum with a given ID. This deletes the forum as well as all topics and posts under it. forumID must be greater than 0 (zero).
	DeleteTopic (see page 889)	Deletes a topic by topicID. This method deletes all posts associated with the topic as well. Returns boolean value.
	GetActiveTopics (see page 890)	This is the overview for the GetActiveTopics method overload.
	GetActiveTopicsForForum (see page 891)	Gets Most active topics for a forum. Activity (see page 996) is measured by number of posts on the topic.
	GetActiveUsers (see page 891)	Gets Most active topics for a board. Activity (see page 996) is measured by number of posts on the topic.
	GetBoardCategories (see page 891)	Method to get the Categories for a board by ID. Returns either zero length or populated array. See DiscussionCategory (see page 1614) class for more information.
	GetBoardForums (see page 892)	Method to get the Forums for a board by ID. Returns either zero length or populated array. See DiscussionForum class for more information.
	GetDiscussionBoard (see page 892)	This is the overview for the GetDiscussionBoard method overload.
	GetDiscussionBoardString (see page 893)	Method to get the rendered view of a discussion board by ID. Returns a string of the output based upon the input parameters. Display of the board can be controlled by using cascading style sheets (CSS).
	GetForum (see page 893)	Method to obtain a particular forum by ID. Returns a discussion board with a 1 length Discussion Forum (see page 264) array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned. *See Also:* The Example section.

	GetRepliesForTopics (see page 895)	Method returns a list of replies for a given forum ID. • ArrayList Structure ArrayListItem(j)(0) • content_id ArrayListItem(j)(1) • comment_total ArrayListItem(j)(2) • comment_sum ArrayListItem(j)(3) • last_posted ArrayListItem(j)(4) • content_language
	GetTopic (see page 895)	Method to obtain a particular Topic by ID. Returns a discussion board with a 1 length Discussion Forum (see page 264) array containing a 1 length DiscussionTopic array. If the topic does not exist, then a DiscussionBoard with a 1 length Forums property containing a zero length Topics property will be returned.
	SearchReplies (see page 896)	Method to obtain an array of topic replies matching search criteria. Returns an instance of EkTasks, which is effectively an array of EkTask objects. If no results are returned, then a zero length EkTasks will be returned.
	SubscribeToBoard (see page 896)	Subscribe a user to a board by providing the ID of the board, the user's ID and the type of subscription.
	SubscribeToForum (see page 897)	Subscribe a user to a forum by providing the ID of the forum, the user's ID and the type of subscription.
	SubscribeToTopic (see page 897)	Subscribe a user to a topic by providing the ID of the topic, the user's ID and the type of subscription.
	UnSubscribeToBoard (see page 897)	Unsubscribe a user from a discussion board by providing the discussion board's ID and user ID.
	UnSubscribeToForum (see page 898)	Unsubscribe a user from a forum by providing the topic ID and user ID.
	UnSubscribeToTopic (see page 898)	Unsubscribe a user from a topic by providing the topic ID and user ID.

ThreadedDiscussion.AddBoard Method

Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns numeric ID of new discussion board.

C#

```
public Long AddBoard(Long ParentID, String Name, String Title, Boolean RequiresAuthentication,
String StyleSheet, String Categories);
```

Visual Basic

```
Public Function AddBoard(ByVal ParentID As Long, ByVal Name As String, ByVal Title As String,
ByVal RequiresAuthentication As Boolean, ByVal StyleSheet As String, ByVal Categories() As
String) As Long
```

Parameters

Parameters	Description
ByVal ParentID As Long	Numeric ID of folder to which you are adding the discussion board.
ByVal Name As String	Name of discussion board.
ByVal Title As String	Title of the discussion board.
ByVal RequiresAuthentication As Boolean	Set to True to require authentication.
ByVal StyleSheet As String	The stylesheet to apply to the discussion board.
ByVal Categories() As String	String array of categories for the discussion board.

Returns

Numeric ID of new discussion board. (Long)

ThreadedDiscussion.AddForum Method

Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns numeric ID of new discussion forum as Long.

See Also: The Example section.

C#

```
public Long AddForum(Long BoardID, String Name, String Description, Boolean ModeratePosts,
Boolean LockForum, int SortOrder, Long CategoryID);
```

Visual Basic

```
Public Function AddForum(ByVal BoardID As Long, ByVal Name As String, ByVal Description As
String, ByVal ModeratePosts As Boolean, ByVal LockForum As Boolean, ByVal SortOrder As
Integer, ByVal CategoryID As Long) As Long
```

Parameters

Parameters	Description
ByVal BoardID As Long	The numeric ID of the board to which the forum will be added.
ByVal Name As String	The string name of the forum being added.
ByVal Description As String	String value for the forum summary.
ByVal ModeratePosts As Boolean	Are posts moderated?
ByVal LockForum As Boolean	Is the forum locked? **True** - prevent posting to the forum. Only members of the Administrators user group can add, edit or delete posts. **False** - the forum is unlocked.
ByVal SortOrder As Integer	Sort order for the new forum. For example, if you have multiple forums, you can assign an integer order to sort them. The order in which the forums appear, correspond to the numbers assigned to each one. 1 (one) is first on the list, 2 (two) appears second and so on. If you assign the same number to more than one forum, those forums are sorted in alphabetical order.
ByVal CategoryID As Long	Numeric ID of the category to which the forum belongs.

Returns

Numeric ID of new discussion forum. (Long)

Example

The following example shows how to create a Web page from which a forum can be added to a discussion board. This example uses some standard drag and drop controls and a small section of VB code utilizing the AddForum method. This method uses the InternalAdmin to add the forum. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. In the Web form, create a table with four rows and three columns.
2. In the first row, first column, add the following label:
```
<asp:Label ID="Label1" runat="server" Text="Board Id: "></asp:Label>
```
3. In the first row, second column, add a text box to enter the board ID.
```
<asp:TextBox ID="txtBoardId" runat="server" Width="63px"></asp:TextBox>
```
4. In the first row, third column, add a RequiredFieldValidator control. This validates that something appears in the board ID box.
```
<asp:RequiredFieldValidator ID="rfvBoardId" runat="server" ControlToValidate="txtBoardId"
  ErrorMessage="Please enter board ID"></asp:RequiredFieldValidator>
```
5. In the second row, first column, add the following label:
```
<asp:Label ID="Label4" runat="server" Text="Forum name: " Width="80px"></asp:Label>
```
6. In the second row, second column, add a text box to enter the forum name.
```
<asp:TextBox ID="txtForumName" runat="server" Width="254px"></asp:TextBox>
```
7. In the second row, third column, add a RequiredFieldValidator control. This validates that something appears in the Forum Name box.
```
<asp:RequiredFieldValidator ID="rfvForumName" runat="server" ErrorMessage="Please enter
forum name"
   ControlToValidate="txtForumName"></asp:RequiredFieldValidator>
```
8. In the third row, first column, add the following label:
```
<asp:Label ID="Label7" runat="server" Text="Description:" Width="76px"></asp:Label>
```
9. In the third row, second column, add a text box to enter a description.
```
<asp:TextBox ID="txtDescription" runat="server" Width="253px"></asp:TextBox>
```
10. In the third row, third column, add a RequiredFieldValidator control. This validates that something appears in the description box.
```
<asp:RequiredFieldValidator ID="rfvDescription" runat="server" ErrorMessage="Please enter
forum description"
```

```
        ControlToValidate="txtDescription" Width="186px"></asp:RequiredFieldValidator>
```

11. In the fourth row, first column, add the following label:

```
    <asp:Label ID="Label2" runat="server" Text="Category ID: " Width="76px"></asp:Label>
```

12. In the fourth row, second column, add a text box to enter a Category ID.

```
    <asp:TextBox ID="txtCategoryId" runat="server" Width="75px"></asp:TextBox>
```

13. In the fourth row, third column, add a RequiredFieldValidator control. This validates that something appears in the Category ID box.

```
    <asp:RequiredFieldValidator ID="rfvCatId" runat="server" ErrorMessage="Please enter
category Id"
        ControlToValidate="txtCategoryId"></asp:RequiredFieldValidator>
```

14. After the table, add a couple of breaks for spacing purposes.

```
    <br /><br />
```

15. Add a button, to submit the information.

```
    <asp:Button ID="btnAddForum" runat="server" Text="Add Forum" Width="72px" />
```

16. Add a couple of more breaks for spacing purposes.

```
    <br /><br />
```

17. Add a results label that will let the person using the page know whether or not the add succeeded.

```
    <asp:Label ID="lblResult" runat="server" Width="549px" Height="25px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
        Protected Sub btnGetForum_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnAddForum.Click
```

2. Create objects for the following items.

```
            Dim board As New Ektron.Cms.DiscussionBoard
            Dim forumApi As New Ektron.Cms.API.Content.ThreadedDiscussion
            Dim forumID As long
```

3. Check to make sure the Board ID is numeric.

```
            If Not IsNumeric(txtBoardId.Text) Then
                lblResult.Text = "Invalid Board ID"
                Exit Sub
            End If
```

4. Check to make sure the Category ID is numeric.

```
            If Not IsNumeric(txtCategoryId.Text) Then
                lblResult.Text = "Invalid Category ID"
                Exit Sub
            End If
```

5. Create a Try/Catch around the AddForum method. This allows you to catch any exceptions and helps when debugging.

```
            Try
                forumID = forumApi.AddForum(CInt(txtBoardId.Text), txtForumName.Text,
txtDescription.Text, False, False, 1, CInt(txtCategoryId.Text))
```

6. Add an If statement that displays a message stating if the forum was added and its ID.

```
                If (forumID <> 0) Then
                    lblResult.Text = "Forum added to the board with Forum ID = " & forumID
                End If
```

7. If there is an error, let the user know what happened.

```
            Catch ex As Exception
                lblResult.Text = ex.Message
            End Try

        End Sub
```

AddReply Method

This is the overview for the AddReply method overload.

Overload List

	Name	Description
	ThreadedDiscussion.AddReply (Long, String, Long, Integer, Boolean, Boolean, String, FileAttachment) (see page 887)	Adds a Post to the Forum (see page 264). You can specify a post to reply to. You can post anonymously.
	ThreadedDiscussion.AddReply (Long, String, Long, Integer, DateTime, Boolean, Boolean, String, FileAttachment) (see page 887)	Adds a post to a topic. Set PostDate property to backdate a post. You can specify a post to reply to. You can post anonymously.

ThreadedDiscussion.AddReply Method (Long, String, Long, Integer, Boolean, Boolean, String, FileAttachment)

Adds a Post to the Forum (see page 264). You can specify a post to reply to. You can post anonymously.

C#

```
public Long AddReply(Long TopicID, String Message, Long ForUserID, int ReplyToPost, Boolean
Approved, Boolean Anonymous, String IPAddress, FileAttachment FileAttachments);
```

Visual Basic

```
Public Function AddReply(ByVal TopicID As Long, ByVal Message As String, ByVal ForUserID As
Long, ByVal ReplyToPost As Integer, ByVal Approved As Boolean, ByVal Anonymous As Boolean,
ByVal IPAddress As String, ByVal FileAttachments() As FileAttachment) As Long
```

Parameters

Parameters	Description
ByVal TopicID As Long	The numeric ID of the topic to add the post to.
ByVal Message As String	The unencoded HTML of the message.
ByVal ForUserID As Long	The ID of the user who is posting.
ByVal ReplyToPost As Integer	The ID of the post to which the user is replying.
ByVal Approved As Boolean	Set to True to approve the post. Otherwise, the post is set to appending.
ByVal Anonymous As Boolean	Set to True to add the post anonymously. Otherwise, the post will be attributed to the logged in user.
ByVal IPAddress As String	IP address of the poster.
ByVal FileAttachments() As FileAttachment	Array of file attachment classes. If attaching documents or images to the post.

Returns

Numeric representing the ID of the post that was added. (Long)

ThreadedDiscussion.AddReply Method (Long, String, Long, Integer, DateTime, Boolean, Boolean, String, FileAttachment)

Adds a post to a topic. Set PostDate property to backdate a post. You can specify a post to reply to. You can post anonymously.

C#

```
public Long AddReply(Long TopicID, String Message, Long ForUserID, int ReplyToPost, DateTime
PostDate, Boolean Approved, Boolean Anonymous, String IPAddress, FileAttachment
FileAttachments);
```

Visual Basic

```
Public Function AddReply(ByVal TopicID As Long, ByVal Message As String, ByVal ForUserID As
Long, ByVal ReplyToPost As Integer, ByVal PostDate As DateTime, ByVal Approved As Boolean,
ByVal Anonymous As Boolean, ByVal IPAddress As String, ByVal FileAttachments() As
```

```
FileAttachment) As Long
```

Parameters

Parameters	Description
ByVal TopicID As Long	The topic to add the post to.
ByVal Message As String	The unencoded HTML of the message.
ByVal ForUserID As Long	The ID of the user who is posting.
ByVal ReplyToPost As Integer	The id of the post to which the user is replying to.
ByVal PostDate As DateTime	DateTime value of when the post occurred.
ByVal Approved As Boolean	Set to True to approve the post. Otherwise, the post is set to appending.
ByVal Anonymous As Boolean	Set to True to add the post anonymously. Otherwise, the post will be attributed to the logged in user.
ByVal IPAddress As String	IP address of the poster.
ByVal FileAttachments() As FileAttachment	Array of file attachment classes. If attaching documents or images to the post.

Returns

Numeric value representing the ID of the post that was added. (Long)

AddTopic Method

This is the overview for the AddTopic method overload.

Description

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as Long. This method is overloaded.

Overload List

	Name	Description
	ThreadedDiscussion.AddTopic (Long, String, String) (see page 888)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as Long.
	ThreadedDiscussion.AddTopic (Long, String, String, Integer) (see page 889)	Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as Long.

ThreadedDiscussion.AddTopic Method (Long, String, String)

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as Long.

C#

```
public Long AddTopic(Long ForumID, String Title, String Message);
```

Visual Basic

```
Public Function AddTopic(ByVal ForumID As Long, ByVal Title As String, ByVal Message As
String) As Long
```

Parameters

Parameters	Description
ByVal ForumID As Long	Numeric ID of forum to which the topic is added.
ByVal Title As String	String value of the topic title.
ByVal Message As String	String value of the topic message.

Returns

Numeric value for the topic ID. (Long)

ThreadedDiscussion.AddTopic Method (Long, String, String, Integer)

Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as Long.

C#

```
public Long AddTopic(Long ForumID, String Title, String Message, int LanguageId);
```

Visual Basic

```
Public Function AddTopic(ByVal ForumID As Long, ByVal Title As String, ByVal Message As
String, ByVal LanguageId As Integer) As Long
```

Parameters

Parameters	Description
ByVal ForumID As Long	Numeric ID of forum to which the topic is added.
ByVal Title As String	String value of the topic title.
ByVal Message As String	String value of the topic message.
ByVal LanguageId As Integer	Numeric ID of the Language.

Returns

Numeric value for the topic ID. (Long)

ThreadedDiscussion.DeleteBoard Method

Deletes a board with a given ID. This deletes the board as well as all forums, topics and posts under it. BoardID must be greater than 0.

C#

```
public DeleteBoard(Long BoardID);
```

Visual Basic

```
Public Sub DeleteBoard(ByVal BoardID As Long)
```

Parameters

Parameters	Description
ByVal BoardID As Long	The numeric ID of the board. This corresponds to the folder ID.

ThreadedDiscussion.DeleteForum Method

Deletes a forum with a given ID. This deletes the forum as well as all topics and posts under it. forumID must be greater than 0 (zero).

C#

```
public DeleteForum(Long forumID);
```

Visual Basic

```
Public Sub DeleteForum(ByVal forumID As Long)
```

Parameters

Parameters	Description
ByVal forumID As Long	The numeric ID of the forum. This corresponds to the folder ID.

ThreadedDiscussion.DeleteTopic Method

Deletes a topic by topicID. This method deletes all posts associated with the topic as well. Returns boolean value.

C#

```
public Boolean DeleteTopic(Long topicid);
```

Visual Basic

```
Public Function DeleteTopic(ByVal topicid As Long) As Boolean
```

Parameters

Parameters	Description
ByVal topicid As Long	Numeric ID of the topic to delete.

Returns

Boolean

GetActiveTopics Method

This is the overview for the GetActiveTopics method overload.

Overload List

	Name	Description
	ThreadedDiscussion.GetActiveTopics (Long, Integer) (see page 890)	Gets Most active topics for a board. Activity (see page 996) is measured by number of posts on the topic.
	ThreadedDiscussion.GetActiveTopics (Long, Integer, TimeSpan) (see page 890)	Gets Most active topics for a board. Activity (see page 996) is measured by number of posts on the topic.

ThreadedDiscussion.GetActiveTopics Method (Long, Integer)

Gets Most active topics for a board. Activity (see page 996) is measured by number of posts on the topic.

C#

```
public DiscussionTopic[] GetActiveTopics(Long boardId, int MaxNumber);
```

Visual Basic

```
Public Function GetActiveTopics(ByVal boardId As Long, ByVal MaxNumber As Integer) As
DiscussionTopic()
```

Parameters

Parameters	Description
ByVal boardId As Long	The numeric ID value of the discussion board.
ByVal MaxNumber As Integer	The maximum number of topics to retrieve.

Returns

An array of Discussion Topic classes.

ThreadedDiscussion.GetActiveTopics Method (Long, Integer, TimeSpan)

Gets Most active topics for a board. Activity (see page 996) is measured by number of posts on the topic.

C#

```
public DiscussionTopic[] GetActiveTopics(Long boardId, int MaxNumber, TimeSpan timeSpan);
```

Visual Basic

```
Public Function GetActiveTopics(ByVal boardId As Long, ByVal MaxNumber As Integer, ByVal
timeSpan As TimeSpan) As DiscussionTopic()
```

Parameters

Parameters	Description
ByVal boardId As Long	The numeric representing the board ID.
ByVal MaxNumber As Integer	The maximum number of topics to retrieve.

Returns

An array of Discussion Topic classes.

ThreadedDiscussion.GetActiveTopicsForForum Method

Gets Most active topics for a forum. Activity (see page 996) is measured by number of posts on the topic.

C#

```
public DiscussionTopic[] GetActiveTopicsForForum(Long forumId, int MaxNumber);
```

Visual Basic

```
Public Function GetActiveTopicsForForum(ByVal forumId As Long, ByVal MaxNumber As Integer) As
DiscussionTopic()
```

Parameters

Parameters	Description
ByVal forumId As Long	The numeric ID value of the forum.
ByVal MaxNumber As Integer	The maximum number of topics to retrieve.

Returns

An array of Discussion Topic classes.

ThreadedDiscussion.GetActiveUsers Method

Gets Most active topics for a board. Activity (see page 996) is measured by number of posts on the topic.

C#

```
public UserData[] GetActiveUsers(Long boardId, int MaxNumber, TimeSpan timeSpan);
```

Visual Basic

```
Public Function GetActiveUsers(ByVal boardId As Long, ByVal MaxNumber As Integer, ByVal
timeSpan As TimeSpan) As UserData()
```

Parameters

Parameters	Description
ByVal boardId As Long	The numeric value representing the board ID.
ByVal MaxNumber As Integer	The maximum number of topics to retrieve.

Returns

An array of Discussion Topic classes.

ThreadedDiscussion.GetBoardCategories Method

Method to get the Categories for a board by ID. Returns either zero length or populated array. See DiscussionCategory (see page 1614) class for more information.

C#

```
public DiscussionCategory[] GetBoardCategories(Long boardID);
```

Visual Basic

```
Public Function GetBoardCategories(ByVal boardID As Long) As DiscussionCategory()
```

Parameters

Parameters	Description
ByVal boardID As Long	Numeric ID of the discussion board.

Returns

Returns an array of type DiscussionCategory (see page 1614).

ThreadedDiscussion.GetBoardForums Method

Method to get the Forums for a board by ID. Returns either zero length or populated array. See DiscussionForum class for more information.

C#

```
public DiscussionForum[] GetBoardForums(Long boardID);
```

Visual Basic

```
Public Function GetBoardForums(ByVal boardID As Long) As DiscussionForum()
```

Parameters

Parameters	Description
ByVal boardID As Long	Numeric value of the board ID.

Returns

Returns an array of type DiscussionForum.

GetDiscussionBoard Method

This is the overview for the GetDiscussionBoard method overload.

Description

Method to get a discussion board by ID or path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

Overload List

	Name	Description
	ThreadedDiscussion.GetDiscussionBoard (Long) (see page 892)	Method to get a board by ID. Returns a DiscussionBoard class for a given ID value. See DiscussionBoard class for more information on the properties of the returned object.
	ThreadedDiscussion.GetDiscussionBoard (String) (see page 893)	Method to get a discussion board by using its path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

ThreadedDiscussion.GetDiscussionBoard Method (Long)

Method to get a board by ID. Returns a DiscussionBoard class for a given ID value. See DiscussionBoard class for more information on the properties of the returned object.

C#

```
public DiscussionBoard GetDiscussionBoard(Long boardID);
```

Visual Basic

```
Public Function GetDiscussionBoard(ByVal boardID As Long) As DiscussionBoard
```

Parameters

Parameters	Description
ByVal boardID As Long	Numeric value of the board ID.

Returns

Returns a class of type DiscussionBoard.

ThreadedDiscussion.GetDiscussionBoard Method (String)

Method to get a discussion board by using its path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

C#

```
public DiscussionBoard GetDiscussionBoard(String boardpath);
```

Visual Basic

```
Public Function GetDiscussionBoard(ByVal boardpath As String) As DiscussionBoard
```

Parameters

Parameters	Description
ByVal boardpath As String	String value of the board path. Ex: EktronCMS400.NET.

Returns

Returns a class of type DiscussionBoard.

ThreadedDiscussion.GetDiscussionBoardString Method

Method to get the rendered view of a discussion board by ID. Returns a string of the output based upon the input parameters. Display of the board can be controlled by using cascading style sheets (CSS).

C#

```
public String GetDiscussionBoardString(Long boardID, String URLpath, System.Web.UI.Page page);
```

Visual Basic

```
Public Function GetDiscussionBoardString(ByVal boardID As Long, ByVal URLpath As String, ByVal
page As System.Web.UI.Page) As String
```

Parameters

Parameters	Description
ByVal boardID As Long	Numeric ID of the board.
ByVal URLpath As String	String value of the URL path used. This is generally the Server Variable "URL". Ex: "/forum.aspx".

Returns

Returns a string of the rendered Discussion Board, suitable for the server control or a Web service.

ThreadedDiscussion.GetForum Method

Method to obtain a particular forum by ID. Returns a discussion board with a 1 length Discussion Forum (see page 264) array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.

See Also: The Example section.

C#

```
public DiscussionBoard GetForum(Long forumid);
```

Visual Basic

```
Public Function GetForum(ByVal forumid As Long) As DiscussionBoard
```

Parameters

Parameters	Description
ByVal forumid As Long	Numeric value of the forum ID.

Returns

An instance of type DiscussionBoard.

Example

The following example shows how to create a Web page from which forum data can be retrieved. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetForum method. This method uses the InternalAdmin to retrieve the forum data. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a label with the following information.
```
<asp:Label ID="Label1" runat="server" Text="Forum ID: "></asp:Label>
```
2. Add a text box to enter the Forum ID.
```
<asp:TextBox ID="txtForumId" runat="server" Width="63px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Forum ID Box.
```
<asp:RequiredFieldValidator ID="rfvForumId" runat="server" ControlToValidate="txtForumId"
   ErrorMessage="Please enter forum ID"></asp:RequiredFieldValidator>
```
4. Add a couple of breaks for spacing purposes.
```
<br /><br />
```
5. Add a button, to retrieve the forum data.
```
<asp:Button ID="btnGetForum" runat="server" Text="Get Forum" Width="72px" />
```
6. Add a couple of more breaks for spacing purposes.
```
<br /><br />
```
7. Add a results label where the forum data will appear. This label is also used to display any errors that might occur.
```
<asp:Label ID="lblResult" runat="server" Width="604px" Height="113px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.
```
    Protected Sub btnGetForum_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGetForum.Click
```
2. Create objects for the following items.
```
Dim board As New Ektron.Cms.DiscussionBoard
Dim forumApi As New Ektron.Cms.API.Content.ThreadedDiscussion
```
3. Check to make sure the forum ID is numeric.
```
If Not IsNumeric(txtForumId.Text) Then
    lblResult.Text = "Invalid Forum ID"
    Exit Sub
End If
```
4. Create a Try/Catch around the GetForum method. This allows you to catch any exceptions and helps when debugging.
```
Try
    board = forumApi.GetForum(CInt(txtForumId.Text))
```
5. Check to make sure the discussion board and forum are valid.
```
    If (board.Id = 0) Or (board.Forums.Length <> 1) Then
        lblResult.Text = "Error getting forum"
        Exit Sub
    End If
```
6. Display the Discussion Board data.

```
                lblResult.Text = "Board Title: " & board.Title & "<br/>"
                lblResult.Text &= "Board Name: " & board.Name & "<br/>"
```

7. Display the Forum (see page 264) data.

```
                lblResult.Text &= "Forum Category: " & board.Forums(0).CategoryName & "<br/>"
                lblResult.Text &= "Forum Name: " & board.Forums(0).ForumName & "<br/>"
                lblResult.Text &= "Forum Title: " & board.Forums(0).ForumTitle & "<br/>"
                lblResult.Text &= "Number of topics: " & board.Forums(0).NumberofTopics &
"<br/>"
                lblResult.Text &= "Number of posts: " & board.Forums(0).NumberofPosts & "<br/>"
```

8. If there is an error, let the user know what happened.

```
            Catch ex As Exception
                lblResult.Text = ex.Message
            End Try

        End Sub
```

ThreadedDiscussion.GetRepliesForTopics Method

Method returns a list of replies for a given forum ID.

- ArrayList Structure ArrayListItem(j)(0)
- content_id ArrayListItem(j)(1)
- comment_total ArrayListItem(j)(2)
- comment_sum ArrayListItem(j)(3)
- last_posted ArrayListItem(j)(4)
- content_language

C#

```
public ArrayList GetRepliesForTopics(Long ForumID);
```

Visual Basic

```
Public Function GetRepliesForTopics(ByVal ForumID As Long) As ArrayList
```

Parameters

Parameters	Description
ByVal ForumID As Long	Numeric value of the forum ID.

Returns

An instance of type ArrayList.

Remarks

This is an older method and is deprecated. The information returned herein was merged into the higher performance GetForum (see page 893)(int forumID) method, and that should be used instead.

ThreadedDiscussion.GetTopic Method

Method to obtain a particular Topic by ID. Returns a discussion board with a 1 length Discussion Forum (see page 264) array containing a 1 length DiscussionTopic array. If the topic does not exist, then a DiscussionBoard with a 1 length Forums property containing a zero length Topics property will be returned.

C#

```
public DiscussionBoard GetTopic(Long topicid);
```

Visual Basic

```
Public Function GetTopic(ByVal topicid As Long) As DiscussionBoard
```

Parameters

Parameters	Description
ByVal topicid As Long	Numeric ID of the Topic.

Returns

An instance of type DiscussionBoard.

ThreadedDiscussion.SearchReplies Method

Method to obtain an array of topic replies matching search criteria. Returns an instance of EkTasks, which is effectively an array of EkTask objects. If no results are returned, then a zero length EkTasks will be returned.

C#

```
public EkTasks SearchReplies(Long ForumIDs, String SearchTerms, EkEnumeration.SearchTypes
stType);
```

Visual Basic

```
Public Function SearchReplies(ByVal ForumIDs() As Long, ByVal SearchTerms() As String, ByVal
stType As EkEnumeration.SearchTypes) As EkTasks
```

Parameters

Parameters	Description
ByVal ForumIDs() As Long	Numeric (Long) array of the Forum (see page 264) IDs you want to search. Use zero length array, or 0, to search all.
ByVal SearchTerms() As String	String array of the search terms (case insensitive).
ByVal stType As EkEnumeration.SearchTypes	EkEnumeration.SearchTypes enumeration value for the search type.

Returns

An instance of type ekTasks (an array of Tasks).

ThreadedDiscussion.SubscribeToBoard Method

Subscribe a user to a board by providing the ID of the board, the user's ID and the type of subscription.

C#

```
public ObjectWatch SubscribeToBoard(Long BoardID, Long ForUserID,
EkEnumeration.NotificationType WatchType);
```

Visual Basic

```
Public Function SubscribeToBoard(ByVal BoardID As Long, ByVal ForUserID As Long, ByVal
WatchType As EkEnumeration.NotificationType) As ObjectWatch
```

Parameters

Parameters	Description
ByVal BoardID As Long	Numeric ID of the discussion board to subscribe to.
ByVal ForUserID As Long	Numeric ID of the user subscribing.
ByVal WatchType As EkEnumeration.NotificationType	Type of subscription. Choices are: • **All** - subscribes a user to all discussion board items. • **Replies** - subscribes a user to all replies to his posts in the specified discussion board. • **Never** - unsubscribes a user from the discussion board.

Returns

Object watch of the current subscription status.

ThreadedDiscussion.SubscribeToForum Method

Subscribe a user to a forum by providing the ID of the forum, the user's ID and the type of subscription.

C#

```
public ObjectWatch SubscribeToForum(Long ForumID, Long ForUserID,
EkEnumeration.NotificationType WatchType);
```

Visual Basic

```
Public Function SubscribeToForum(ByVal ForumID As Long, ByVal ForUserID As Long, ByVal
WatchType As EkEnumeration.NotificationType) As ObjectWatch
```

Parameters

Parameters	Description
ByVal ForumID As Long	Numeric ID of the forum to subscribe to.
ByVal ForUserID As Long	Numeric ID of the user subscribing.
ByVal WatchType As EkEnumeration.NotificationType	Type of subscription. Choices are: • **All** - subscribes a user to all forum items. • **Replies** - subscribes a user to all replies to his posts in the specified forum. • **Never** - unsubscribes a user from the forum.

Returns

Object watch of the current subscription status.

ThreadedDiscussion.SubscribeToTopic Method

Subscribe a user to a topic by providing the ID of the topic, the user's ID and the type of subscription.

C#

```
public ObjectWatch SubscribeToTopic(Long TopicID, Long ForUserID,
EkEnumeration.NotificationType WatchType);
```

Visual Basic

```
Public Function SubscribeToTopic(ByVal TopicID As Long, ByVal ForUserID As Long, ByVal
WatchType As EkEnumeration.NotificationType) As ObjectWatch
```

Parameters

Parameters	Description
ByVal TopicID As Long	Numeric ID of the topic to subscribe to.
ByVal ForUserID As Long	Numeric ID of the user subscribing.
ByVal WatchType As EkEnumeration.NotificationType	Type of subscription. Choices are: • **All** - subscribes a user to all topics. • **Replies** - subscribes a user to any replies to the topic he created. • **Never** - unsubscribes a user from the topic.

Returns

Object watch of the current subscription status.

ThreadedDiscussion.UnSubscribeToBoard Method

Unsubscribe a user from a discussion board by providing the discussion board's ID and user ID.

C#

```
public ObjectWatch UnSubscribeToBoard(Long BoardID, Long ForUserID);
```

Visual Basic

```
Public Function UnSubscribeToBoard(ByVal BoardID As Long, ByVal ForUserID As Long) As
ObjectWatch
```

Parameters

Parameters	Description
ByVal BoardID As Long	Numeric ID of the discussion board to unsubscribe from.
ByVal ForUserID As Long	Numeric ID of the user to unsubscribe.

Returns

Object watch of the current subscription status.

ThreadedDiscussion.UnSubscribeToForum Method

Unsubscribe a user from a forum by providing the topic ID and user ID.

C#

```
public ObjectWatch UnSubscribeToForum(Long ForumID, Long ForUserID);
```

Visual Basic

```
Public Function UnSubscribeToForum(ByVal ForumID As Long, ByVal ForUserID As Long) As
ObjectWatch
```

Parameters

Parameters	Description
ByVal ForumID As Long	Numeric ID of the forum to unsubscribe from.
ByVal ForUserID As Long	Numeric ID of the user to unsubscribe.

Returns

Object watch of the current subscription status.

ThreadedDiscussion.UnSubscribeToTopic Method

Unsubscribe a user from a topic by providing the topic ID and user ID.

C#

```
public ObjectWatch UnSubscribeToTopic(Long TopicID, Long ForUserID);
```

Visual Basic

```
Public Function UnSubscribeToTopic(ByVal TopicID As Long, ByVal ForUserID As Long) As
ObjectWatch
```

Parameters

Parameters	Description
ByVal TopicID As Long	Numeric ID of the topic to unsubscribe from.
ByVal ForUserID As Long	Numeric ID of the user unsubscribing.

Returns

Object watch of the current subscription status.

Ektron.Cms.API.Search Namespace

The Search Developer API (see page 669) enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 725), if you chose to install it, has excellent examples of programmatically using the SearchManager (see page 899). It is installed at http://~YourSite~/CMS400Developer/Websearch/WebsearchApi.aspx. Examples are in VB and C#.

Here is a list of some of the examples.

- Retrieve a document or HTML content that contains the word *cms*.
- Retrieve content that contains *research* and has a searchable metadata *wellness* set to Cancer.
- Retrieve images that contain *RotatingGraphic1* in their title.
- Retrieve geo-content from CMS400.Net.

Classes

	Name	Description
	SearchManager (see page 899)	The Search (see page 903) Developer API (see page 669) enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 725), if you chose to install it, has excellent examples of programmatically using the SearchManager. It is installed at http://~YourSite~/CMS400Developer/Developer/Websearch/WebsearchApi.aspx. Examples are in VB and C#. Here is a list of some of the examples. • Retrieve a document or HTML content that contains the word *cms*. • Retrieve content that contains *research* and has a searchable metadata *wellness* set to Cancer. • Retrieve images that contain *RotatingGraphic1* in their title. • Retrieve geo-content from CMS400.Net.

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	SearchManager (see page 899)	The Search (see page 903) Developer API (see page 669) enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 725), if you chose to install it, has excellent examples of programmatically using the SearchManager. It is installed at http://~YourSite~/CMS400Developer/Developer/Websearch/WebsearchApi.aspx. Examples are in VB and C#. Here is a list of some of the examples. • Retrieve a document or HTML content that contains the word *cms*. • Retrieve content that contains *research* and has a searchable metadata *wellness* set to Cancer. • Retrieve images that contain *RotatingGraphic1* in their title. • Retrieve geo-content from CMS400.Net.

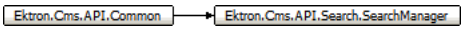
SearchManager Class

The Search (see page 903) Developer API (see page 669) enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 725), if you chose to install it, has excellent examples of programmatically using the SearchManager. It is installed at http://~YourSite~/CMS400Developer/Developer/Websearch/WebsearchApi.aspx. Examples are in VB and C#.

Here is a list of some of the examples.

- Retrieve a document or HTML content that contains the word *cms*.
- Retrieve content that contains *research* and has a searchable metadata *wellness* set to Cancer.
- Retrieve images that contain *RotatingGraphic1* in their title.
- Retrieve geo-content from CMS400.Net.

Class Hierarchy



C#

```
public class SearchManager : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class SearchManager
Inherits Ektron.Cms.API.Common
```

File

SearchManager.vb

SearchManager Methods

The methods of the SearchManager class are listed here.

Public Methods

	Name	Description
	Execute (see page 900)	This is the overview for the Execute method overload.
	Search (see page 902)	This is the overview for the Search method overload.
	SearchForCategories (see page 906)	This is the overview for the SearchForCategories method overload.
	SearchTable (see page 907)	This is the overview for the SearchTable method overload.

Execute Method

This is the overview for the Execute method overload.

Description

Returns the array of ContentData. This method is overloaded.

Overload List

	Name	Description
	SearchManager.Execute (Ektron.Cms.ContentSearchCondition) (see page 900)	Returns the array of ContentData based on the criteria ContentSearchCondition.
	SearchManager.Execute (Ektron.Cms.UserSearchCondition) (see page 901)	Returns the array of UserData based on the criteria UserSearchCondition.

SearchManager.Execute Method (Ektron.Cms.ContentSearchCondition)

Returns the array of ContentData based on the criteria ContentSearchCondition.

C#

```
public Overridable ContentData[] Execute(Ektron.Cms.ContentSearchCondition condition);
```

Visual Basic

```
Public Overridable Function Execute(ByVal condition As Ektron.Cms.ContentSearchCondition) As
ContentData()
```

Parameters

Parameters	Description
ByVal condition As Ektron.Cms.ContentSearchCondition	ContentSearchCondition

Returns

Array of ContentData

Example

```
Dim isContentLanguage As ContentSearchCondition = New ContentSearchCondition
isContentLanguage.setType = SearchType.EQUAL
isContentLanguage.setValue = 1033 'The value can be integer,string,date and boolean should be
match with db type
isContentLanguage.setVariable = content.content_language

Dim isInEktron As ContentSearchCondition = New ContentSearchCondition
isInEktron.setType = SearchType.EQUAL
isInEktron.setValue = "03031"  'The value can be integer,string,date and boolean should be
match with dynamic_data_tbl labels type
isInEktron.setVariable = "customproperties.zip code"

Dim condition As ContentSearchCondition = New ContentSearchCondition
condition.setType = SearchType.AND
condition.AddCondition(isInEktron)
condition.AddCondition(isMemberShip)

Dim search As New SearchManager
Dim result As userdata() = search.Execute(condition)
```

SearchManager.Execute Method (Ektron.Cms.UserSearchCondition)

Returns the array of UserData based on the criteria UserSearchCondition.

C#

```
public Overridable UserData[] Execute(Ektron.Cms.UserSearchCondition condition);
```

Visual Basic

```
Public Overridable Function Execute(ByVal condition As Ektron.Cms.UserSearchCondition) As
UserData()
```

Parameters

Parameters	Description
ByVal condition As Ektron.Cms.UserSearchCondition	UserSearchCondition

Returns

Array of UserData

Example

```
Dim isMemberShip As UserSearchCondition = New UserSearchCondition
isMemberShip.setType = SearchType.EQUAL
isMemberShip.setValue = 1 'The value can be integer,string,date and boolean should be match
with db type
isMemberShip.setVariable = users.membership_user

Dim isInEktron As UserSearchCondition = New UserSearchCondition
isInEktron.setType = SearchType.EQUAL
```

```
isInEktron.setValue = "03031"  'The value can be integer,string,date and boolean should be
match with dynamic_data_tbl labels type
isInEktron.setVariable = "customproperties.zip code"

Dim condition As UserSearchCondition = New UserSearchCondition
condition.setType = SearchType.AND
condition.AddCondition(isInEktron)
condition.AddCondition(isMemberShip)

Dim search As New SearchManager
Dim result As userdata() = search.Execute(condition)
```

Search Method

This is the overview for the Search method overload.

Overload List

	Name	Description
	SearchManager.Search (IndexSearchRequestData, HttpContext, Integer) (see page 903)	Search smart form content based on indexed fields. The IndexSearchFieldList property (an array of IndexSearchFieldData) of the IndexSearchRequestData object is used to specify the XPATH and their values.
	SearchManager.Search (MapRequestData, HttpContext, Integer, Integer, Integer) (see page 903)	Map (see page 350) search for retrieving geographical content from CMS400.Net site. This function has two additional overloads.
	SearchManager.Search (MapRequestData, HttpContext, Integer, Integer, Integer, DataTable, DataTable) (see page 903)	Map (see page 350) search for retrieving geographical content from CMS400.Net site. This function has two additional overloads.
	SearchManager.Search (SearchRequestData, HttpContext, Integer) (see page 904)	Search HTML, content, images, multimedia and documents based on content or metadata. This function has two additional overloads.
	SearchManager.Search (SearchRequestData, HttpContext, Integer, DataTable) (see page 904)	Search HTML, content, images, multimedia and documents based on content or metadata. This function has two additional overloads.
	SearchManager.Search (SearchRequestData, HttpContext, Integer, DataTable, DataTable) (see page 905)	Search HTML, content, images, multimedia and documents based on content or metadata. This function has two additional overloads.
	SearchManager.Search (String, HttpContext, Integer) (see page 905)	Search the site for occurrence(s) of search text. This function has an additional overload.
	SearchManager.Search (String, HttpContext, Integer, DataTable, DataTable) (see page 906)	Search the site for occurrence(s) of search text. This function has an additional overload.

SearchManager.Search Method (IndexSearchRequestData, HttpContext, Integer)

Search smart form content based on indexed fields.

The IndexSearchFieldList property (an array of IndexSearchFieldData) of the IndexSearchRequestData object is used to specify the XPATH and their values.

C#

```
public SearchResponseData[] Search(IndexSearchRequestData indexSearchReqData, HttpContext
context, ref int resultCount);
```

Visual Basic

```
Public Function Search(ByVal indexSearchReqData As IndexSearchRequestData, ByVal context As
HttpContext, ByRef resultCount As Integer) As SearchResponseData()
```

Parameters

Parameters	Description
ByVal indexSearchReqData As IndexSearchRequestData	Search criteria bundled in a IndexSearchRequestData object.
ByVal context As HttpContext	The current Http request.
ByRef resultCount As Integer	Number of records returned.

Returns

Array of SearchResponseData - content satisfying search criteria

SearchManager.Search Method (MapRequestData, HttpContext, Integer, Integer, Integer)

Map (see page 350) search for retrieving geographical content from CMS400.Net site.

This function has two additional overloads.

C#

```
public MapResponseData[] Search(MapRequestData mapReqData, HttpContext context, int index, int
count, ref int resultCount);
```

Visual Basic

```
Public Function Search(ByVal mapReqData As MapRequestData, ByVal context As HttpContext, ByVal
index As Integer, ByVal count As Integer, ByRef resultCount As Integer) As MapResponseData()
```

Parameters

Parameters	Description
ByVal mapReqData As MapRequestData	Search criteria bundled in a MapRequestData object.
ByVal context As HttpContext	The current Http request.
ByVal index As Integer	Page number.
ByVal count As Integer	Page size.
ByRef resultCount As Integer	Number of records returned.

Returns

Array of MapResponseData - content satisfying location criteria

SearchManager.Search Method (MapRequestData, HttpContext, Integer, Integer, Integer, DataTable, DataTable)

Map (see page 350) search for retrieving geographical content from CMS400.Net site.

This function has two additional overloads.

C#

```
public MapResponseData[] Search(MapRequestData mapReqData, HttpContext context, int index, int count, ref int resultCount, ref DataTable suggestedResultTable, ref DataTable privateResultTable);
```

Visual Basic

```
Public Function Search(ByVal mapReqData As MapRequestData, ByVal context As HttpContext, ByVal index As Integer, ByVal count As Integer, ByRef resultCount As Integer, ByRef suggestedResultTable As DataTable, ByRef privateResultTable As DataTable) As MapResponseData()
```

Parameters

Parameters	Description
ByVal mapReqData As MapRequestData	Search criteria bundled in a MapRequestData object.
ByVal context As HttpContext	The current Http request.
ByVal index As Integer	Page number.
ByVal count As Integer	Page size.
ByRef resultCount As Integer	Number of records returned.

Returns

Array of MapResponseData - content satisfying location criteria

SearchManager.Search Method (SearchRequestData, HttpContext, Integer)

Search HTML, content, images, multimedia and documents based on content or metadata.

This function has two additional overloads.

C#

```
public SearchResponseData[] Search(SearchRequestData searchReqData, HttpContext context, ref int resultCount);
```

Visual Basic

```
Public Function Search(ByVal searchReqData As SearchRequestData, ByVal context As HttpContext, ByRef resultCount As Integer) As SearchResponseData()
```

Parameters

Parameters	Description
ByVal searchReqData As SearchRequestData	Search criteria bundled in a SearchRequestData object.
ByVal context As HttpContext	The current Http request.
ByRef resultCount As Integer	Number of records returned.

Returns

Array of SearchResponseData - content satisfying search criteria

SearchManager.Search Method (SearchRequestData, HttpContext, Integer, DataTable)

Search HTML, content, images, multimedia and documents based on content or metadata.

This function has two additional overloads.

C#

```
public SearchResponseData[] Search(SearchRequestData searchReqData, HttpContext context, ref int resultCount, ref DataTable suggestedResultTable);
```

Visual Basic

```
Public Function Search(ByVal searchReqData As SearchRequestData, ByVal context As HttpContext, ByRef resultCount As Integer, ByRef suggestedResultTable As DataTable) As SearchResponseData()
```

Parameters

Parameters	Description
ByVal searchReqData As SearchRequestData	Search criteria bundled in a SearchRequestData object.
ByVal context As HttpContext	The current Http request.
ByRef resultCount As Integer	Number of records returned.
ByRef suggestedResultTable As DataTable	Datatable passed by reference in which suggested results are returned.

Returns

Array of SearchResponseData - content satisfying search criteria

SearchManager.Search Method (SearchRequestData, HttpContext, Integer, DataTable, DataTable)

Search HTML, content, images, multimedia and documents based on content or metadata.

This function has two additional overloads.

C#

```
public SearchResponseData[] Search(SearchRequestData searchReqData, HttpContext context, ref
int resultCount, ref DataTable suggestedResultTable, ref DataTable privateResultTable);
```

Visual Basic

```
Public Function Search(ByVal searchReqData As SearchRequestData, ByVal context As HttpContext,
ByRef resultCount As Integer, ByRef suggestedResultTable As DataTable, ByRef
privateResultTable As DataTable) As SearchResponseData()
```

Parameters

Parameters	Description
ByVal searchReqData As SearchRequestData	Search criteria bundled in a SearchRequestData object.
ByVal context As HttpContext	The current Http request.
ByRef resultCount As Integer	Number of records returned.
ByRef suggestedResultTable As DataTable	Datatable passed by reference in which suggested results are returned.
ByRef privateResultTable As DataTable	Datatable passed by reference in which private results are returned.

Returns

Array of SearchResponseData - content satisfying search criteria

SearchManager.Search Method (String, HttpContext, Integer)

Search the site for occurrence(s) of search text.

This function has an additional overload.

C#

```
public SearchResponseData[] Search(String searchText, HttpContext context, ref int
resultCount);
```

Visual Basic

```
Public Function Search(ByVal searchText As String, ByVal context As HttpContext, ByRef
resultCount As Integer) As SearchResponseData()
```

Parameters

Parameters	Description
ByVal searchText As String	The text for which you are searching.
ByVal context As HttpContext	The current Http request.
ByRef resultCount As Integer	Number of records returned.

Returns

Array of SearchResponseData - content satisfying search criteria

SearchManager.Search Method (String, HttpContext, Integer, DataTable, DataTable)

Search the site for occurrence(s) of search text.

This function has an additional overload.

C#

```
public SearchResponseData[] Search(String searchText, HttpContext context, ref int
resultCount, ref DataTable suggestedResultTable, ref DataTable privateResultTable);
```

Visual Basic

```
Public Function Search(ByVal searchText As String, ByVal context As HttpContext, ByRef
resultCount As Integer, ByRef suggestedResultTable As DataTable, ByRef privateResultTable As
DataTable) As SearchResponseData()
```

Parameters

Parameters	Description
ByVal searchText As String	The text for which you are searching.
ByVal context As HttpContext	The current Http request.
ByRef resultCount As Integer	Number of records returned.
ByRef suggestedResultTable As DataTable	Datatable passed by reference in which suggested results are returned.
ByRef privateResultTable As DataTable	Datatable passed by reference in which private results are returned.

Returns

Array of SearchResponseData - content satisfying search criteria

SearchForCategories Method

This is the overview for the SearchForCategories method overload.

Overload List

	Name	Description
	SearchManager.SearchForCategories (SearchRequestData, HttpContext, Integer, Integer, Integer) (see page 906)	Search (see page 903) based on taxonomy category.
	SearchManager.SearchForCategories (SearchRequestData, HttpContext, Integer, Integer, Integer, DataTable, DataTable) (see page 907)	Search (see page 903) based on taxonomy category.

SearchManager.SearchForCategories Method (SearchRequestData, HttpContext, Integer, Integer, Integer)

Search (see page 903) based on taxonomy category.

C#

```
public String[] SearchForCategories(SearchRequestData searchReqData, HttpContext context, int
index, int count, ref int resultCount);
```

Visual Basic

```
Public Function SearchForCategories(ByVal searchReqData As SearchRequestData, ByVal context As
HttpContext, ByVal index As Integer, ByVal count As Integer, ByRef resultCount As Integer) As
```

```
String()
```

Parameters

Parameters	Description
ByVal searchReqData As SearchRequestData	Search (see page 903) criteria bundled in a SearchRequestData object.
ByVal context As HttpContext	The current Http request.
ByVal index As Integer	Page number.
ByVal count As Integer	Page size.
ByRef resultCount As Integer	Number of records returned.

Returns

Array of String - List of categories

SearchManager.SearchForCategories Method (SearchRequestData, HttpContext, Integer, Integer, Integer, DataTable, DataTable)

Search (see page 903) based on taxonomy category.

C#

```
public String[] SearchForCategories(SearchRequestData searchReqData, HttpContext context, int
index, int count, ref int resultCount, ref DataTable suggestedResultTable, ref DataTable
privateResultTable);
```

Visual Basic

```
Public Function SearchForCategories(ByVal searchReqData As SearchRequestData, ByVal context As
HttpContext, ByVal index As Integer, ByVal count As Integer, ByRef resultCount As Integer,
ByRef suggestedResultTable As DataTable, ByRef privateResultTable As DataTable) As String()
```

Parameters

Parameters	Description
ByVal searchReqData As SearchRequestData	Search (see page 903) criteria bundled in a SearchRequestData object.
ByVal context As HttpContext	The current Http request.
ByVal index As Integer	Page number.
ByVal count As Integer	Page size.
ByRef resultCount As Integer	Number of records returned.
ByRef suggestedResultTable As DataTable	Datatable passed by reference in which suggested results are returned.
ByRef privateResultTable As DataTable	Datatable passed by reference in which private results are returned.

Returns

Array of String - List of categories

SearchTable Method

This is the overview for the SearchTable method overload.

Overload List

	Name	Description
	SearchManager.SearchTable (SearchRequestData, String, HttpContext, Integer) (see page 908)	This API (see page 669) is used internally by the Ektron Windows Service to update catalogs when a taxonomy category is updated This ensures that subsequent searches function correctly.
	SearchManager.SearchTable (SearchRequestData, String, HttpContext, Integer, DataTable, DataTable) (see page 908)	This API (see page 669) is used internally by the Ektron Windows Service to update catalogs when a taxonomy category is updated This ensures that subsequent searches function correctly.

SearchManager.SearchTable Method (SearchRequestData, String, HttpContext, Integer)

This API (see page 669) is used internally by the Ektron Windows Service to update catalogs when a taxonomy category is updated

This ensures that subsequent searches function correctly.

C#

```
public DataSet SearchTable(SearchRequestData searchReqData, String searchParams, HttpContext
context, ref int resultCount);
```

Visual Basic

```
Public Function SearchTable(ByVal searchReqData As SearchRequestData, ByVal searchParams As
String, ByVal context As HttpContext, ByRef resultCount As Integer) As DataSet
```

Parameters

Parameters	Description
ByVal searchReqData As SearchRequestData	Search (see page 903) criteria bundled in a SearchRequestData object.
ByVal searchParams As String	Database columns to be included in search.
ByVal context As HttpContext	The current Http request.
ByRef resultCount As Integer	Number of records returned.

Returns

DataSet with a single DataTable which contains content whose taxonomy was updated.

SearchManager.SearchTable Method (SearchRequestData, String, HttpContext, Integer, DataTable, DataTable)

This API (see page 669) is used internally by the Ektron Windows Service to update catalogs when a taxonomy category is updated

This ensures that subsequent searches function correctly.

C#

```
public DataSet SearchTable(SearchRequestData searchReqData, String searchParams, HttpContext
context, ref int resultCount, ref DataTable suggestedResultTable, ref DataTable
privateResultTable);
```

Visual Basic

```
Public Function SearchTable(ByVal searchReqData As SearchRequestData, ByVal searchParams As
String, ByVal context As HttpContext, ByRef resultCount As Integer, ByRef suggestedResultTable
As DataTable, ByRef privateResultTable As DataTable) As DataSet
```

Parameters

Parameters	Description
ByVal searchReqData As SearchRequestData	Search (see page 903) criteria bundled in a SearchRequestData object.
ByVal searchParams As String	Database columns to be included in search.
ByVal context As HttpContext	The current Http request.
ByRef resultCount As Integer	Number of records returned.
ByRef suggestedResultTable As DataTable	Datatable passed by reference in which suggested results are returned.
ByRef privateResultTable As DataTable	Datatable passed by reference in which private results are returned.

Returns

DataSet with a single DataTable which contains content whose taxonomy was updated.

Ektron.Cms.API.Task Namespace

This is namespace Ektron.Cms.API.Task.

Classes

	Name	Description
	Task (see page 909)	This is class Ektron.Cms.API.Task.Task.
	TaskCategory (see page 919)	This is class Ektron.Cms.API.Task.TaskCategory.
	TaskCategoryType (see page 922)	This is class Ektron.Cms.API.Task.TaskCategoryType.

Classes

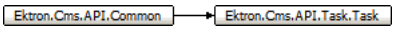
The following table lists classes in this documentation.

Classes

	Name	Description
	Task (see page 909)	This is class Ektron.Cms.API.Task.Task.
	TaskCategory (see page 919)	This is class Ektron.Cms.API.Task.TaskCategory.
	TaskCategoryType (see page 922)	This is class Ektron.Cms.API.Task.TaskCategoryType.

Task Class

Class Hierarchy



C#

```
public class Task : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Task
Inherits Ektron.Cms.API.Common
```

File

Task.vb

Description

This is class Ektron.Cms.API.Task.Task.

Task Methods

The methods of the Task class are listed here.

Public Methods

	Name	Description
	AddTask (see page 910)	Adds a task in the system.
	AssignTaskToUser (see page 910)	This is the overview for the AssignTaskToUser method overload.
	AssignTaskToUsergroup (see page 911)	This is the overview for the AssignTaskToUsergroup method overload.
	DeleteTask (see page 912)	This is the overview for the DeleteTask method overload.

	DeleteTasks (see page 913)	This is the overview for the DeleteTasks method overload.
	GetTask (see page 914)	Get the task information.
	GetTasks (see page 914)	This is the overview for the GetTasks method overload.
	GetTasksAssignedByUser (see page 915)	This is the overview for the GetTasksAssignedByUser method overload.
	GetTasksAssignedToUser (see page 916)	This is the overview for the GetTasksAssignedToUser method overload.
	GetTasksAssignedToUsergroup (see page 917)	This is the overview for the GetTasksAssignedToUsergroup method overload.
	GetTaskState (see page 918)	Get the task state.
	SetStateForTask (see page 919)	Set the task state.
	UpdateTask (see page 919)	Updates the task

Task.AddTask Method

Adds a task in the system.

C#

```
public Boolean AddTask(ref TaskData data);
```

Visual Basic

```
Public Function AddTask(ByRef data As TaskData) As Boolean
```

Parameters

Parameters	Description
ByRef data As TaskData	TaskData

Returns

Returns False if successfully added, otherwise returns True.

AssignTaskToUser Method

This is the overview for the AssignTaskToUser method overload.

Overload List

	Name	Description
	Task.AssignTaskToUser (Long, Long) (see page 910)	Assign a task to a user.
	Task.AssignTaskToUser (Long, Long, Long) (see page 911)	Assign a task to a user by a specified user.

Task.AssignTaskToUser Method (Long, Long)

Assign a task to a user.

C#

```
public Boolean AssignTaskToUser(Long taskId, Long assignToUserId);
```

Visual Basic

```
Public Function AssignTaskToUser(ByVal taskId As Long, ByVal assignToUserId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal taskId As Long	Task (see page 909) ID
ByVal assignToUserId As Long	User (see page 950)'s ID

Returns

Returns False if the task is successfully assigned to a user. Otherwise, it returns True.

Task.AssignTaskToUser Method (Long, Long, Long)

Assign a task to a user by a specified user.

C#

```
public Boolean AssignTaskToUser(Long taskId, Long assignToUserId, Long assignByUserId);
```

Visual Basic

```
Public Function AssignTaskToUser(ByVal taskId As Long, ByVal assignToUserId As Long, ByVal
assignByUserId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal taskId As Long	Task (see page 909) ID
ByVal assignToUserId As Long	Assign to User (see page 950)'s ID
ByVal assignByUserId As Long	Assign by User (see page 950)'s ID

Returns

Returns False if the task is successfully assigned to a user. Otherwise, it returns True.

AssignTaskToUsergroup Method

This is the overview for the AssignTaskToUsergroup method overload.

Overload List

	Name	Description
	Task.AssignTaskToUsergroup (Long, Long) (see page 911)	Assign a task to a usergroup.
	Task.AssignTaskToUsergroup (Long, Long, Long) (see page 912)	Assign a task to a usergroup by a specified user.

Task.AssignTaskToUsergroup Method (Long, Long)

Assign a task to a usergroup.

C#

```
public Boolean AssignTaskToUsergroup(Long taskId, Long assignToUsergroup);
```

Visual Basic

```
Public Function AssignTaskToUsergroup(ByVal taskId As Long, ByVal assignToUsergroup As Long)
As Boolean
```

Parameters

Parameters	Description
ByVal taskId As Long	Task (see page 909) ID

ByVal assignToUsergroup As Long	Assign to Usergroup's ID

Returns

Returns False if the task is successfully assigned to a usergroup. Otherwise, it returns True.

Task.AssignTaskToUsergroup Method (Long, Long, Long)

Assign a task to a usergroup by a specified user.

C#

```
public Boolean AssignTaskToUsergroup(Long taskId, Long assignToUsergroup, Long assignByUserId);
```

Visual Basic

```
Public Function AssignTaskToUsergroup(ByVal taskId As Long, ByVal assignToUsergroup As Long,
ByVal assignByUserId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal taskId As Long	Task (see page 909) ID
ByVal assignToUsergroup As Long	Numeric ID of the Usergroup to which the task is being assigned.
ByVal assignByUserId As Long	Numeric ID of the User (see page 950) who's assigning the task.

Returns

Returns False if the task is successfully assigned to a usergroup. Otherwise, it returns True.

DeleteTask Method

This is the overview for the DeleteTask method overload.

Overload List

	Name	Description
	Task.DeleteTask (Long) (see page 912)	Mark the task deleted.
	Task.DeleteTask (Long, Boolean) (see page 913)	Mark the task deleted or delete the task permanently.

Task.DeleteTask Method (Long)

Mark the task deleted.

C#

```
public Boolean DeleteTask(Long taskId);
```

Visual Basic

```
Public Function DeleteTask(ByVal taskId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal taskId As Long	Task (see page 909) ID.

Returns

Returns False if successfully deleted, otherwise returns True.

Task.DeleteTask Method (Long, Boolean)

Mark the task deleted or delete the task permanently.

C#

```
public Boolean DeleteTask(Long taskId, Boolean Permanently);
```

Visual Basic

```
Public Function DeleteTask(ByVal taskId As Long, ByVal Permanently As Boolean) As Boolean
```

Parameters

Parameters	Description
ByVal taskId As Long	Task (see page 909) ID.
ByVal Permanently As Boolean	True=Delete the task.

Returns

Returns False if successfully deleted otherwise returns True.

DeleteTasks Method

This is the overview for the DeleteTasks method overload.

Overload List

	Name	Description
	Task.DeleteTasks (Long) (see page 913)	Mark tasks deleted.
	Task.DeleteTasks (Long, Boolean) (see page 913)	Mark tasks deleted or delete tasks permanently.

Task.DeleteTasks Method (Long)

Mark tasks deleted.

C#

```
public Boolean DeleteTasks(Long taskId);
```

Visual Basic

```
Public Function DeleteTasks(ByVal taskId() As Long) As Boolean
```

Parameters

Parameters	Description
ByVal taskId() As Long	Array of Task (see page 909) ID.

Returns

Returns False if successfully deleted, otherwise returns True.

Remarks

When an error occurs the method will quit without deleting rest of the tasks.

Task.DeleteTasks Method (Long, Boolean)

Mark tasks deleted or delete tasks permanently.

C#

```
public Boolean DeleteTasks(Long taskId, Boolean Permanently);
```

Visual Basic

```
Public Function DeleteTasks(ByVal taskId() As Long, ByVal Permanently As Boolean) As Boolean
```

Parameters

Parameters	Description
ByVal taskId() As Long	Array of Task (see page 909) ID.
ByVal Permanently As Boolean	True=Delete tasks.

Returns

Returns False if successfully deleted, otherwise returns True.

Remarks

When an error occurs the method will quit without deleting rest of the tasks.

Task.GetTask Method

Get the task information.

C#

```
public TaskData GetTask(Long taskId);
```

Visual Basic

```
Public Function GetTask(ByVal taskId As Long) As TaskData
```

Parameters

Parameters	Description
ByVal taskId As Long	The task's numeric ID.

Returns

TaskData

GetTasks Method

This is the overview for the GetTasks method overload.

Overload List

	Name	Description
	Task.GetTasks () (see page 914)	Get All tasks from the system.
	Task.GetTasks (Long) (see page 915)	Get All tasks from the system.

Task.GetTasks Method ()

Get All tasks from the system.

C#

```
public TaskData[] GetTasks();
```

Visual Basic

```
Public Function GetTasks() As TaskData()
```

Returns

Array of task data.

Task.GetTasks Method (Long)

Get All tasks from the system.

C#

```
public TaskData[] GetTasks(Long folderId);
```

Visual Basic

```
Public Function GetTasks(ByVal folderId() As Long) As TaskData()
```

Parameters

Parameters	Description
ByVal folderId() As Long	Array of FolderIDs

Returns

Array of task data.

GetTasksAssignedByUser Method

This is the overview for the GetTasksAssignedByUser method overload.

Overload List

	Name	Description
	Task.GetTasksAssignedByUser (Long) (see page 915)	Get all tasks assigned by the user.
	Task.GetTasksAssignedByUser (Long, EkEnumeration.TaskState) (see page 915)	Get all tasks assign by a user with a specified task state and returns them as array of TaskData.

Task.GetTasksAssignedByUser Method (Long)

Get all tasks assigned by the user.

C#

```
public TaskData[] GetTasksAssignedByUser(Long userId);
```

Visual Basic

```
Public Function GetTasksAssignedByUser(ByVal userId As Long) As TaskData()
```

Parameters

Parameters	Description
ByVal userId As Long	Numeric value of the user's ID.

Returns

Array of tasks.

Task.GetTasksAssignedByUser Method (Long, EkEnumeration.TaskState)

Get all tasks assign by a user with a specified task state and returns them as array of TaskData.

C#

```
public TaskData[] GetTasksAssignedByUser(Long userId, EkEnumeration.TaskState taskState);
```

Visual Basic

```
Public Function GetTasksAssignedByUser(ByVal userId As Long, ByVal taskState As
EkEnumeration.TaskState) As TaskData()
```

Parameters

Parameters	Description
ByVal userId As Long	Numeric value of the user's ID.
ByVal taskState As EkEnumeration.TaskState	The state of the task. Choices are: • NotStarted • Active • AwaitingData • OnHold • Pending • Reopened • Completed • Archived • Deleted

Returns

Array of tasks.

Description

GetTasksAssignedToUser Method

This is the overview for the GetTasksAssignedToUser method overload.

Overload List

	Name	Description
	Task.GetTasksAssignedToUser (Long) (see page 916)	Get all tasks assigned to the user and returns them as array of TaskData.
	Task.GetTasksAssignedToUser (Long, EkEnumeration.TaskState) (see page 917)	Get all tasks assigned to the user by a specified task state and returns them as array of TaskData.

Task.GetTasksAssignedToUser Method (Long)

Get all tasks assigned to the user and returns them as array of TaskData.

C#

```
public TaskData[] GetTasksAssignedToUser(Long userId);
```

Visual Basic

```
Public Function GetTasksAssignedToUser(ByVal userId As Long) As TaskData()
```

Parameters

Parameters	Description
ByVal userId As Long	The numeric value of the User (see page 950) ID.

Returns

Returns an array of tasks assigned to the user.

Task.GetTasksAssignedToUser Method (Long, EkEnumeration.TaskState)

Get all tasks assigned to the user by a specified task state and returns them as array of TaskData.

C#

```
public TaskData[] GetTasksAssignedToUser(Long userId, EkEnumeration.TaskState taskState);
```

Visual Basic

```
Public Function GetTasksAssignedToUser(ByVal userId As Long, ByVal taskState As
EkEnumeration.TaskState) As TaskData()
```

Parameters

Parameters	Description
ByVal userId As Long	Numeric value of the User (see page 950)'s ID.
ByVal taskState As EkEnumeration.TaskState	The state of the task. Choices are: • NotStarted • Active • AwaitingData • OnHold • Pending • Reopened • Completed • Archived • Deleted

Returns

Returns an array of TaskData.

Description

GetTasksAssignedToUsergroup Method

This is the overview for the GetTasksAssignedToUsergroup method overload.

Overload List

	Name	Description
	Task.GetTasksAssignedToUsergroup (Long) (see page 918)	Get all tasks assigned to the usergroup and returns them as array of TaskData.
	Task.GetTasksAssignedToUsergroup (Long, EkEnumeration.TaskState) (see page 918)	Get all tasks assigned to the usergroup by a specified task state and returns them as array of TaskData.

Task.GetTasksAssignedToUsergroup Method (Long)

Get all tasks assigned to the usergroup and returns them as array of TaskData.

C#

```
public TaskData[] GetTasksAssignedToUsergroup(Long groupId);
```

Visual Basic

```
Public Function GetTasksAssignedToUsergroup(ByVal groupId As Long) As TaskData()
```

Parameters

Parameters	Description
ByVal groupId As Long	The numeric value of the usergroup's ID.

Returns

Returns an array of tasks.

Task.GetTasksAssignedToUsergroup Method (Long, EkEnumeration.TaskState)

Get all tasks assigned to the usergroup by a specified task state and returns them as array of TaskData.

C#

```
public TaskData[] GetTasksAssignedToUsergroup(Long groupId, EkEnumeration.TaskState taskState);
```

Visual Basic

```
Public Function GetTasksAssignedToUsergroup(ByVal groupId As Long, ByVal taskState As
EkEnumeration.TaskState) As TaskData()
```

Parameters

Parameters	Description
ByVal groupId As Long	Numeric value of the Usergroup's ID.
ByVal taskState As EkEnumeration.TaskState	The state of the task. Choices are: • NotStarted • Active • AwaitingData • OnHold • Pending • Reopened • Completed • Archived • Deleted

Returns

Returns an array of tasks.

Description

Task.GetTaskState Method

Get the task state.

C#

```
public EkEnumeration.TaskState GetTaskState(Long taskId);
```

Visual Basic

```
Public Function GetTaskState(ByVal taskId As Long) As EkEnumeration.TaskState
```

Parameters

Parameters	Description
ByVal taskId As Long	Task (see page 909) ID.

Returns

If successful returns task state. Otherwise, it will throw an error.

Task.SetStateForTask Method

Set the task state.

C#

```
public Boolean SetStateForTask(Long taskId, EkEnumeration.TaskState newTaskState);
```

Visual Basic

```
Public Function SetStateForTask(ByVal taskId As Long, ByVal newTaskState As
EkEnumeration.TaskState) As Boolean
```

Parameters

Parameters	Description
ByVal taskId As Long	Task (see page 909) ID
ByVal newTaskState As EkEnumeration.TaskState	New Task (see page 909) State

Returns

Returns False if the state is successfully changed. Otherwise, it returns True.

Task.UpdateTask Method

Updates the task

C#

```
public Boolean UpdateTask(ref TaskData data);
```

Visual Basic

```
Public Function UpdateTask(ByRef data As TaskData) As Boolean
```

Parameters

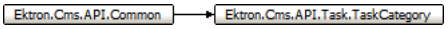
Parameters	Description
ByRef data As TaskData	Task (see page 909) Data.

Returns

Returns False if successfully updated, otherwise returns True.

TaskCategory Class

Class Hierarchy



C#

```
public class TaskCategory : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class TaskCategory
Inherits Ektron.Cms.API.Common
```

File

TaskCategory.vb

Description

This is class Ektron.Cms.API.Task.TaskCategory.

TaskCategory Methods

The methods of the TaskCategory class are listed here.

Public Methods

	Name	Description
	AddTaskCategory (see page 920)	Adds a task category by providing a title.
	DeleteTaskCategory (see page 920)	Deletes the task category by providing its numeric ID.
	GetAllCategories (see page 921)	Gets All Task (see page 909) categories and all of the task types that belong to those categories.
	GetTaskCategory (see page 921)	Gets the specified task category and all the task types that belong to it by providing the numeric ID of the category.
	UpdateTaskCategory (see page 921)	Updates the category title by providing its ID and a new title.

TaskCategory.AddTaskCategory Method

Adds a task category by providing a title.

C#

```
public Long AddTaskCategory(String title);
```

Visual Basic

```
Public Function AddTaskCategory(ByVal title As String) As Long
```

Parameters

Parameters	Description
ByVal title As String	The new category title.

Returns

Returns a numeric ID assigned to the new task Category. The data type is Long.

TaskCategory.DeleteTaskCategory Method

Deletes the task category by providing its numeric ID.

C#

```
public Boolean DeleteTaskCategory(Long categoryId);
```

Visual Basic

```
Public Function DeleteTaskCategory(ByVal categoryId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal categoryId As Long	The numeric ID of the task category.

Returns

Returns False if successfully deleted. Otherwise, it returns True.

TaskCategory.GetAllCategories Method

Gets All Task (see page 909) categories and all of the task types that belong to those categories.

C#

```
public TaskCategoryData[] GetAllCategories();
```

Visual Basic

```
Public Function GetAllCategories() As TaskCategoryData()
```

Returns

Returns an array of TaskCategoryData

TaskCategory.GetTaskCategory Method

Gets the specified task category and all the task types that belong to it by providing the numeric ID of the category.

C#

```
public TaskCategoryData GetTaskCategory(Long categoryID);
```

Visual Basic

```
Public Function GetTaskCategory(ByVal categoryID As Long) As TaskCategoryData
```

Parameters

Parameters	Description
ByVal categoryID As Long	The numeric ID of the category.

Returns

Returns TaskCategoryData.

TaskCategory.UpdateTaskCategory Method

Updates the category title by providing its ID and a new title.

C#

```
public Boolean UpdateTaskCategory(Long categoryId, String title);
```

Visual Basic

```
Public Function UpdateTaskCategory(ByVal categoryId As Long, ByVal title As String) As Boolean
```

Parameters

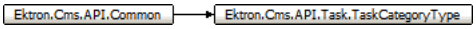
Parameters	Description
ByVal categoryId As Long	The numeric ID of the Category.
ByVal title As String	The new title of the category.

Returns

Returns False if successfully updated. Otherwise, it returns True.

TaskCategoryType Class

Class Hierarchy



C#

```
public class TaskCategoryType : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class TaskCategoryType
Inherits Ektron.Cms.API.Common
```

File

TaskCategoryType.vb

Description

This is class Ektron.Cms.API.Task.TaskCategoryType.

TaskCategoryType Methods

The methods of the TaskCategoryType class are listed here.

Public Methods

	Name	Description
	AddTaskType (see page 922)	Adds a new task type to a category by providing the new task type's title and description, the category's ID and whether to make the type active. Once the new task type has been added, an new ID is returned.
	DeleteTaskType (see page 923)	Deletes the task type by providing its ID.
	GetTaskType (see page 923)	Gets the task type information for a specified ID.
	UpdateTaskType (see page 923)	Updates task type information by providing its ID. You can update the Title, Description and set whether the type is active or not.

TaskCategoryType.AddTaskType Method

Adds a new task type to a category by providing the new task type's title and description, the category's ID and whether to make the type active. Once the new task type has been added, an new ID is returned.

C#

```
public Long AddTaskType(String title, String description, Long categoryId, Boolean active);
```

Visual Basic

```
Public Function AddTaskType(ByVal title As String, ByVal description As String, ByVal
categoryId As Long, ByVal active As Boolean) As Long
```

Parameters

Parameters	Description
ByVal title As String	A title for the new task type.
ByVal description As String	A description for the new task type.
ByVal categoryId As Long	The ID of the category with which the new task type will be associated.
ByVal active As Boolean	Set to True to make the new task type active. Otherwise, set to false.

Returns

Returns the task type's numeric ID.

TaskCategoryType.DeleteTaskType Method

Deletes the task type by providing its ID.

C#

```
public Boolean DeleteTaskType(Long taskTypeId);
```

Visual Basic

```
Public Function DeleteTaskType(ByVal taskTypeId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal taskTypeId As Long	The task type numeric ID to be deleted.

Returns

Returns False if the task type is successfully deleted. Otherwise, it returns True.

TaskCategoryType.GetTaskType Method

Gets the task type information for a specified ID.

C#

```
public TaskTypeData GetTaskType(Long taskTypeId);
```

Visual Basic

```
Public Function GetTaskType(ByVal taskTypeId As Long) As TaskTypeData
```

Parameters

Parameters	Description
ByVal taskTypeId As Long	The numeric ID of the task type.

Returns

Returns TaskTypeData information.

TaskCategoryType.UpdateTaskType Method

Updates task type information by providing its ID. You can update the Title, Description and set whether the type is active or not.

C#

```
public Boolean UpdateTaskType(Long taskTypeId, String title, String description, Boolean
active);
```

Visual Basic

```
Public Function UpdateTaskType(ByVal taskTypeId As Long, ByVal title As String, ByVal
description As String, ByVal active As Boolean) As Boolean
```

Parameters

Parameters	Description
ByVal taskTypeId As Long	The numeric ID of the task type.
ByVal title As String	The title of the task type.
ByVal description As String	A description for the task type.
ByVal active As Boolean	Set to True to make the task type active. Otherwise, set to false.

Returns

Returns False if the task type is successfully updated. Otherwise, it returns True.

Ektron.Cms.API.UrlAliasing Namespace

This is namespace Ektron.Cms.API.UrlAliasing.

Classes

	Name	Description
	UrlAliasAuto (see page 924)	This class allows you to work with automatic aliasing.
	UrlAliasCommon (see page 931)	This class allows you work with the common components of URL Aliasing.
	UrlAliasManual (see page 933)	This class allows you to work with manual aliasing.
	UrlAliasRegEx (see page 941)	This class allows you to work with the Regular Expressions within aliasing.
	UrlAliasSettings (see page 944)	This class allows you to work with settings for enabling and disabling the various types of aliasing available in CMS400.NET. In addition, you can work with alias settings for caching, tracing and page extensions. These settings reflect the same settings in the **Workarea** > **Settings** > **Configuration** > **URL Aliasing** > **Settings** screen.

Classes

The following table lists classes in this documentation.

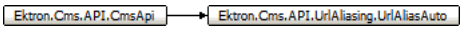
Classes

	Name	Description
	UrlAliasAuto (see page 924)	This class allows you to work with automatic aliasing.
	UrlAliasCommon (see page 931)	This class allows you work with the common components of URL Aliasing.
	UrlAliasManual (see page 933)	This class allows you to work with manual aliasing.
	UrlAliasRegEx (see page 941)	This class allows you to work with the Regular Expressions within aliasing.
	UrlAliasSettings (see page 944)	This class allows you to work with settings for enabling and disabling the various types of aliasing available in CMS400.NET. In addition, you can work with alias settings for caching, tracing and page extensions. These settings reflect the same settings in the **Workarea** > **Settings** > **Configuration** > **URL Aliasing** > **Settings** screen.

UrlAliasAuto Class

This class allows you to work with automatic aliasing.

Class Hierarchy



C#

```
public class UrlAliasAuto : Ektron.Cms.API.CmsApi;
```

Visual Basic

```
Public Class UrlAliasAuto
Inherits Ektron.Cms.API.CmsApi
```

File

UrlAliasAuto.cs

UrlAliasAuto Methods

The methods of the UrlAliasAuto class are listed here.

Public Methods

	Name	Description
	Add (see page 925)	Adds a new automatic alias configuration by providing information for the UrlAliasAutoData (see page 1592) data class.
	ClearCache (see page 925)	Clears the auto alias cache.
	Delete (see page 926)	Deletes an automatic aliasing configuration by providing its ID.
	GetAutoAliasNameTypes (see page 926)	Gets a list of page names for automatic alias types. Types include: • ContentTitle = 1 • ContentId = 2 • ContentIdAndLanguage = 3
	GetAutoAliasTypes (see page 926)	Gets a list of automatic aliasing types. Types include: • Taxonomy (see page 876) = 1 • Folder (see page 683) = 2
	GetFolderItem (see page 926)	Gets a content's single record for folder automatic aliases by providing a content's ID and language.
	GetFolderList (see page 927)	Gets a content's list of records for folder automatic aliases by providing a content's ID and language..
	GetFolderSource (see page 927)	Gets a list of all folders within a site that are avaliable for automatic alising.
	GetItem (see page 927)	Gets a single record for a given automatic aliasing configuration ID.
	GetItemForContent (see page 928)	Gets the first record in the list of automatic aliases for a given content ID. The first record is returned for both folder aliases and taxonomy aliases.
	GetList (see page 928)	Gets a list of records.
	GetListForContent (see page 929)	Gets a list of automatic aliases for a given content ID. The list includes both folder aliases and taxonomy aliases.
	GetTarget (see page 929)	This is the overview for the GetTarget method overload.
	GetTaxonomyItem (see page 930)	Gets a content's single record for taxonomy automatic aliases by providing a content's ID and language.
	GetTaxonomyList (see page 931)	Gets a specified content's list of records for taxonomy automatic aliases by providing a content's ID and language.
	GetTaxonomySource (see page 931)	Gets a list of all taxonomies for a specified language that are avaliable for automatic aliasing.
	Update (see page 931)	Modifies an existing automatic alias configuration by providing information for the UrlAliasAutoData (see page 1592) data class.

UrlAliasAuto.Add Method

Adds a new automatic alias configuration by providing information for the UrlAliasAutoData (see page 1592) data class.

C#

```
public UrlAliasAutoData Add(UrlAliasAutoData data);
```

Visual Basic

```
Public Function Add(data As UrlAliasAutoData) As UrlAliasAutoData
```

Parameters

Parameters	Description
UrlAliasAutoData data	UrlAliasAutoData (see page 1592) information.

UrlAliasAuto.ClearCache Method

Clears the auto alias cache.

C#

```
public void ClearCache();
```

Visual Basic

```
Public Function ClearCache() As void
```

UrlAliasAuto.Delete Method

Deletes an automatic aliasing configuration by providing its ID.

C#

```
public void Delete(long autoId);
```

Visual Basic

```
Public Function Delete(autoId As long) As void
```

Parameters

Parameters	Description
long autoId	The ID of the automatic aliasing configuration.
callerUserId	The ID of the user deleting the automatic aliasing.

UrlAliasAuto.GetAutoAliasNameTypes Method

Gets a list of page names for automatic alias types. Types include:

- ContentTitle = 1
- ContentId = 2
- ContentIdAndLanguage = 3

C#

```
public List<EkEnumeration.AutoAliasNameType> GetAutoAliasNameTypes();
```

Visual Basic

```
Public Function GetAutoAliasNameTypes() As List<EkEnumeration.AutoAliasNameType>
```

Returns

A list of page names for automatic alias types.

UrlAliasAuto.GetAutoAliasTypes Method

Gets a list of automatic aliasing types. Types include:

- Taxonomy (see page 876) = 1
- Folder (see page 683) = 2

C#

```
public List<EkEnumeration.AutoAliasType> GetAutoAliasTypes();
```

Visual Basic

```
Public Function GetAutoAliasTypes() As List<EkEnumeration.AutoAliasType>
```

Returns

A list of automatic aliasing types.

UrlAliasAuto.GetFolderItem Method

Gets a content's single record for folder automatic aliases by providing a content's ID and language.

C#

```
public string GetFolderItem(long contentId, int contentLanguage);
```

Visual Basic

```
Public Function GetFolderItem(contentId As long, contentLanguage As Integer) As string
```

Parameters

Parameters	Description
long contentId	The content ID.
int contentLanguage	The content's language ID.

Returns

A content's single record for folder automatic aliases. Returns an empty record if no data is found.

UrlAliasAuto.GetFolderList Method

Gets a content's list of records for folder automatic aliases by providing a content's ID and language..

C#

```
public List<UrlAliasAutoData> GetFolderList(long contentId, int contentLanguage);
```

Visual Basic

```
Public Function GetFolderList(contentId As long, contentLanguage As Integer) As
List<UrlAliasAutoData>
```

Parameters

Parameters	Description
long contentId	The content ID.
int contentLanguage	The content's language ID.

Returns

A content's list of records for folder automatic aliases. Returns an empty list if no data is found.

UrlAliasAuto.GetFolderSource Method

Gets a list of all folders within a site that are avaliable for automatic alising.

C#

```
public List<FolderData> GetFolderSource(long siteId);
```

Visual Basic

```
Public Function GetFolderSource(siteId As long) As List<FolderData>
```

Parameters

Parameters	Description
long siteId	The ID of the site from which to retrieve a list of folders.

Returns

A list of all folders within a site that are avaliable for automatic alising.

UrlAliasAuto.GetItem Method

Gets a single record for a given automatic aliasing configuration ID.

C#

```
public UrlAliasAutoData GetItem(long autoId);
```

Visual Basic

```
Public Function GetItem(autoId As long) As UrlAliasAutoData
```

Parameters

Parameters	Description
long autoId	The ID of the automatic aliasing configuration.

Returns

A single record for a given automatic aliasing configuration ID. Returns an empty record if no data is found.

UrlAliasAuto.GetItemForContent Method

Gets the first record in the list of automatic aliases for a given content ID. The first record is returned for both folder aliases and taxonomy aliases.

C#

```
public string GetItemForContent(long contentId);
```

Visual Basic

```
Public Function GetItemForContent(contentId As long) As string
```

Parameters

Parameters	Description
long contentId	The content ID.

Returns

The first record in the list of automatic aliases for a given content ID.

UrlAliasAuto.GetList Method

Gets a list of records.

C#

```
public List<UrlAliasAutoData> GetList(PagingInfo paging, long siteId, int contentLanguage,
EkEnumeration.AutoAliasSearchField searchField, string searchValue,
EkEnumeration.AutoAliasOrderBy orderBy);
```

Visual Basic

```
Public Function GetList(paging As PagingInfo, siteId As long, contentLanguage As Integer,
searchField As EkEnumeration.AutoAliasSearchField, searchValue As string, orderBy As
EkEnumeration.AutoAliasOrderBy) As List<UrlAliasAutoData>
```

Parameters

Parameters	Description
PagingInfo paging	The paging object.
long siteId	The Site (see page 725) ID.
int contentLanguage	The language ID.
EkEnumeration.AutoAliasSearchField searchField	The type of search to perform. • None • All • ContentId • ContentTitle • Target • [Alias]
string searchValue	The value for which to search.

EkEnumeration.AutoAliasOrderBy orderBy	Select the order of the list. Choices are: • None • ContentId • ContentLanguage • ContentTitle • IsDefault • IsEnabled • Target • SiteId • QueryString • FileExtension • DisplayAlias

Returns

Gets a list of records. Returns an empty list if no data is found.

UrlAliasAuto.GetListForContent Method

Gets a list of automatic aliases for a given content ID. The list includes both folder aliases and taxonomy aliases.

C#

```
public List<UrlAliasAutoData> GetListForContent(long contentId);
```

Visual Basic

```
Public Function GetListForContent(contentId As long) As List<UrlAliasAutoData>
```

Parameters

Parameters	Description
long contentId	The ID of the content for which to retrieve a list of aliases.

Returns

A list of automatic aliases for a given content ID.

GetTarget Method

This is the overview for the GetTarget method overload.

Overload List

	Name	Description
	UrlAliasAuto.GetTarget (string, string) (see page 930)	Get the destination for a alias name.
	UrlAliasAuto.GetTarget (string, string, EkEnumeration.AutoAliasType) (see page 930)	Get the destination for a alias name for a perticular source.

UrlAliasAuto.GetTarget Method (string, string)

Get the destination for a alias name.

C#

```
public string GetTarget(string urlAlias, string host);
```

Visual Basic

```
Public Function GetTarget(urlAlias As string, host As string) As string
```

Parameters

Parameters	Description
string urlAlias	The alias name without the QueryString.
string host	The host name.

Returns

The destination for a alias name. Returns an empty string ID if no destination is found.

UrlAliasAuto.GetTarget Method (string, string, EkEnumeration.AutoAliasType)

Get the destination for a alias name for a perticular source.

C#

```
public string GetTarget(string urlAlias, string host, EkEnumeration.AutoAliasType
autoAliasType);
```

Visual Basic

```
Public Function GetTarget(urlAlias As string, host As string, autoAliasType As
EkEnumeration.AutoAliasType) As string
```

Parameters

Parameters	Description
string urlAlias	The alias name without the query string.
string host	The host name.
EkEnumeration.AutoAliasType autoAliasType	The source type. Choices are: • Taxonomy (see page 876) • Folder (see page 683)

Returns

The destination for a alias name for a perticular source.

UrlAliasAuto.GetTaxonomyItem Method

Gets a content's single record for taxonomy automatic aliases by providing a content's ID and language.

C#

```
public string GetTaxonomyItem(long contentId, int contentLanguage);
```

Visual Basic

```
Public Function GetTaxonomyItem(contentId As long, contentLanguage As Integer) As string
```

Parameters

Parameters	Description
long contentId	The content ID.
int contentLanguage	The content's language ID.

Returns

A content's single record for taxonomy automatic aliases.

UrlAliasAuto.GetTaxonomyList Method

Gets a specified content's list of records for taxonomy automatic aliases by providing a content's ID and language.

C#

```
public List<UrlAliasAutoData> GetTaxonomyList(long contentId, int contentLanguage);
```

Visual Basic

```
Public Function GetTaxonomyList(contentId As long, contentLanguage As Integer) As
List<UrlAliasAutoData>
```

Parameters

Parameters	Description
long contentId	The content ID.
int contentLanguage	The content's language ID.

UrlAliasAuto.GetTaxonomySource Method

Gets a list of all taxonomies for a specified language that are avaliable for automatic aliasing.

C#

```
public List<TaxonomyData> GetTaxonomySource(int languageId);
```

Visual Basic

```
Public Function GetTaxonomySource(languageId As Integer) As List<TaxonomyData>
```

Parameters

Parameters	Description
int languageId	The taxonomy's language ID.

Returns

A list of all taxonomies for a specified language that are avaliable for automatic aliasing.

UrlAliasAuto.Update Method

Modifies an existing automatic alias configuration by providing information for the UrlAliasAutoData (see page 1592) data class.

C#

```
public UrlAliasAutoData Update(UrlAliasAutoData data);
```

Visual Basic

```
Public Function Update(data As UrlAliasAutoData) As UrlAliasAutoData
```

Parameters

Parameters	Description
UrlAliasAutoData data	UrlAliasAutoData (see page 1592) information.

UrlAliasCommon Class

This class allows you work with the common components of URL Aliasing.

Class Hierarchy

C#

```
public class UrlAliasCommon : Ektron.Cms.API.CmsApi;
```

Visual Basic

```
Public Class UrlAliasCommon
Inherits Ektron.Cms.API.CmsApi
```

File

UrlAliasCommon.cs

UrlAliasCommon Methods

The methods of the UrlAliasCommon class are listed here.

Public Methods

	Name	Description
	GetAliasForContent (see page 932)	Gets all the aliases for a content item.
	GetTarget (see page 932)	This is the overview for the GetTarget method overload.

UrlAliasCommon.GetAliasForContent Method

Gets all the aliases for a content item.

C#

```
public string GetAliasForContent(long contentId);
```

Visual Basic

```
Public Function GetAliasForContent(contentId As long) As string
```

Parameters

Parameters	Description
long contentId	The content's ID.

GetTarget Method

This is the overview for the GetTarget method overload.

Overload List

	Name	Description
	UrlAliasCommon.GetTarget (System.Web.HttpContext) (see page 932)	Get the destination link (partial URL) for an alias.
	UrlAliasCommon.GetTarget (Uri) (see page 933)	Get the destination link (partial URL) for an alias.

UrlAliasCommon.GetTarget Method (System.Web.HttpContext)

Get the destination link (partial URL) for an alias.

C#

```
public string GetTarget(System.Web.HttpContext context);
```

Visual Basic

```
Public Function GetTarget(context As System.Web.HttpContext) As string
```

Parameters

Parameters	Description
url	The Uri object.

UrlAliasCommon.GetTarget Method (Uri)

Get the destination link (partial URL) for an alias.

C#

```
public string GetTarget(Uri url);
```

Visual Basic

```
Public Function GetTarget(url As Uri) As string
```

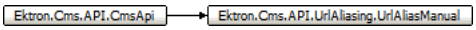
Parameters

Parameters	Description
Uri url	The Uri object.

UrlAliasManual Class

This class allows you to work with manual aliasing.

Class Hierarchy



C#

```
public class UrlAliasManual : Ektron.Cms.API.CmsApi;
```

Visual Basic

```
Public Class UrlAliasManual
Inherits Ektron.Cms.API.CmsApi
```

File

UrlAliasManual.cs

UrlAliasManual Methods

The methods of the UrlAliasManual class are listed here.

Public Methods

	Name	Description
	Add (see page 934)	Adds a new manual alias by providing information for the UrlAliasManualData (see page 1599) data class.
	ClearCache (see page 934)	Clears the cache.
	Delete (see page 934)	Deletes an alias by providing its ID.
	GetDefaultAlias (see page 934)	Gets the primary alias for a given content ID.
	GetFileExtensions (see page 935)	Gets a list of file extensions.
	GetItem (see page 935)	Gets a single record for an alias ID.
	GetList (see page 935)	This is the overview for the GetList method overload.

	GetSiteList (see page 940)	Gets a list of sites.
	GetTemplateList (see page 940)	Gets a list of templates for a given content item by providing the content's ID.
	Update (see page 940)	Update's an existing manual alias by providing information for the UrlAliasManualData (see page 1599) data class.
	ValidateName (see page 940)	Validates the Url Alais.

UrlAliasManual.Add Method

Adds a new manual alias by providing information for the UrlAliasManualData (see page 1599) data class.

C#

```
public UrlAliasManualData Add(UrlAliasManualData data);
```

Visual Basic

```
Public Function Add(data As UrlAliasManualData) As UrlAliasManualData
```

Parameters

Parameters	Description
UrlAliasManualData data	UrlAliasManualData (see page 1599) data.
callerUserId	The ID of the user addding the new manual alias.

Returns

Returns the new record.

UrlAliasManual.ClearCache Method

Clears the cache.

C#

```
public void ClearCache();
```

Visual Basic

```
Public Function ClearCache() As void
```

UrlAliasManual.Delete Method

Deletes an alias by providing its ID.

C#

```
public void Delete(long aliasId);
```

Visual Basic

```
Public Function Delete(aliasId As long) As void
```

Parameters

Parameters	Description
long aliasId	The ID of the alias to be deleted.

UrlAliasManual.GetDefaultAlias Method

Gets the primary alias for a given content ID.

C#

```
public UrlAliasManualData GetDefaultAlias(long contentId);
```

Visual Basic

```
Public Function GetDefaultAlias(contentId As long) As UrlAliasManualData
```

Parameters

Parameters	Description
long contentId	The content's ID.

Returns

The primary alias for a given content ID.

UrlAliasManual.GetFileExtensions Method

Gets a list of file extensions.

C#

```
public List<string> GetFileExtensions();
```

Visual Basic

```
Public Function GetFileExtensions() As List<string>
```

Returns

A list of file extensions.

UrlAliasManual.GetItem Method

Gets a single record for an alias ID.

C#

```
public UrlAliasManualData GetItem(long aliasId);
```

Visual Basic

```
Public Function GetItem(aliasId As long) As UrlAliasManualData
```

Parameters

Parameters	Description
long aliasId	The ID of the alias.

Returns

A single record for an alias ID.

GetList Method

This is the overview for the GetList method overload.

Overload List

	Name	Description
	UrlAliasManual.GetList (PagingInfo, EkEnumeration.UrlAliasingOrderBy) (see page 936)	Gets a list of all the manual aliases.
	UrlAliasManual.GetList (PagingInfo, long, bool, EkEnumeration.UrlAliasingOrderBy) (see page 937)	Gets a list of active aliases or a list of all aliases for a content item by providing its ID.

	UrlAliasManual.GetList (PagingInfo, long, EkEnumeration.UrlAliasingOrderBy) (see page 936)	Gets a list of manual aliases for a site by providing the site's ID.
	UrlAliasManual.GetList (PagingInfo, long, int, EkEnumeration.UrlAliasingOrderBy) (see page 939)	Gets a list of aliases for a site by providing paging, site ID, language and order by information.
	UrlAliasManual.GetList (PagingInfo, long, int, EkEnumeration.UrlAliasSearchField, string, EkEnumeration.UrlAliasingOrderBy) (see page 938)	Gets a list of aliases for a site based on specified search criteria by providing paging, site ID, language, search criteria and order by information.

UrlAliasManual.GetList Method (PagingInfo, EkEnumeration.UrlAliasingOrderBy)

Gets a list of all the manual aliases.

C#

```
public List<UrlAliasManualData> GetList(PagingInfo paging, EkEnumeration.UrlAliasingOrderBy
orderBy);
```

Visual Basic

```
Public Function GetList(paging As PagingInfo, orderBy As EkEnumeration.UrlAliasingOrderBy) As
List<UrlAliasManualData>
```

Parameters

Parameters	Description
PagingInfo paging	The paging object.
EkEnumeration.UrlAliasingOrderBy orderBy	Select the order of the list. Choices are: • None • ContentId • ContentLanguage • ContentTitle • IsDefault • IsEnabled • Target • SiteId • QueryString • FileExtension • DisplayAlias

Returns

A list of all the manual aliases.

UrlAliasManual.GetList Method (PagingInfo, long, EkEnumeration.UrlAliasingOrderBy)

Gets a list of manual aliases for a site by providing the site's ID.

C#

```
public List<UrlAliasManualData> GetList(PagingInfo paging, long siteId,
EkEnumeration.UrlAliasingOrderBy orderBy);
```

Visual Basic

```
Public Function GetList(paging As PagingInfo, siteId As long, orderBy As
EkEnumeration.UrlAliasingOrderBy) As List<UrlAliasManualData>
```

Parameters

Parameters	Description
PagingInfo paging	The paging object.
long siteId	The ID of the site.
EkEnumeration.UrlAliasingOrderBy orderBy	Select the order of the list. Choices are: • None • ContentLanguage • SourceName • Active • Type

Returns

A list of manual aliases for a site.

UrlAliasManual.GetList Method (PagingInfo, long, bool, EkEnumeration.UrlAliasingOrderBy)

Gets a list of active aliases or a list of all aliases for a content item by providing its ID.

C#

```
public List<UrlAliasManualData> GetList(PagingInfo paging, long contentId, bool onlyActive,
EkEnumeration.UrlAliasingOrderBy orderBy);
```

Visual Basic

```
Public Function GetList(paging As PagingInfo, contentId As long, onlyActive As bool, orderBy
As EkEnumeration.UrlAliasingOrderBy) As List<UrlAliasManualData>
```

Parameters

Parameters	Description
PagingInfo paging	The paging object.
long contentId	The content's ID.
bool onlyActive	Set to true to only return a list of active manual aliases.

EkEnumeration.UrlAliasingOrderBy orderBy	Select the order of the list. Choices are: • None • ContentId • ContentLanguage • ContentTitle • IsDefault • IsEnabled • Target • SiteId • QueryString • FileExtension • DisplayAlias

Returns

A list of only active or inactive aliases for a content item.

UrlAliasManual.GetList Method (PagingInfo, long, int, EkEnumeration.UrlAliasSearchField, string, EkEnumeration.UrlAliasingOrderBy)

Gets a list of aliases for a site based on specified search criteria by providing paging, site ID, language, search criteria and order by information.

C#

```
public List<UrlAliasManualData> GetList(PagingInfo paging, long siteId, int contentLanguage,
EkEnumeration.UrlAliasSearchField searchField, string searchValue,
EkEnumeration.UrlAliasingOrderBy orderBy);
```

Visual Basic

```
Public Function GetList(paging As PagingInfo, siteId As long, contentLanguage As Integer,
searchField As EkEnumeration.UrlAliasSearchField, searchValue As string, orderBy As
EkEnumeration.UrlAliasingOrderBy) As List<UrlAliasManualData>
```

Parameters

Parameters	Description
PagingInfo paging	The paging object.
long siteId	The ID of the Site (see page 725).
int contentLanguage	The language ID.
EkEnumeration.UrlAliasSearchField searchField	The type of search to perform. • None • All • ContentId • ContentTitle • Target • [Alias]
string searchValue	The value for which to search.

EkEnumeration.UrlAliasingOrderBy orderBy	Select the order of the list. Choices are: • None • ContentId • ContentLanguage • ContentTitle • IsDefault • IsEnabled • Target • SiteId • QueryString • FileExtension • DisplayAlias

Returns

A list of aliases for a site based on specified search criteria.

UrlAliasManual.GetList Method (PagingInfo, long, int, EkEnumeration.UrlAliasingOrderBy)

Gets a list of aliases for a site by providing paging, site ID, language and order by information.

C#

```
public List<UrlAliasManualData> GetList(PagingInfo paging, long siteId, int contentLanguage,
EkEnumeration.UrlAliasingOrderBy orderBy);
```

Visual Basic

```
Public Function GetList(paging As PagingInfo, siteId As long, contentLanguage As Integer,
orderBy As EkEnumeration.UrlAliasingOrderBy) As List<UrlAliasManualData>
```

Parameters

Parameters	Description
PagingInfo paging	The paging object.
long siteId	The ID of the site.
int contentLanguage	The language ID.
EkEnumeration.UrlAliasingOrderBy orderBy	Select the order of the list. Choices are: • None • ContentId • ContentLanguage • ContentTitle • IsDefault • IsEnabled • Target • SiteId • QueryString • FileExtension • DisplayAlias

Returns

A list of aliases for a site.

UrlAliasManual.GetSiteList Method

Gets a list of sites.

C#

```
public Dictionary<long,string> GetSiteList();
```

Visual Basic

```
Public Function GetSiteList() As Dictionary<long,string>
```

Returns

A list of sites.

UrlAliasManual.GetTemplateList Method

Gets a list of templates for a given content item by providing the content's ID.

C#

```
public List<LibraryData> GetTemplateList(long contentId);
```

Visual Basic

```
Public Function GetTemplateList(contentId As long) As List<LibraryData>
```

Parameters

Parameters	Description
long contentId	The ID of the content.

Returns

A list of templates for a given content item.

UrlAliasManual.Update Method

Update's an existing manual alias by providing information for the UrlAliasManualData (see page 1599) data class.

C#

```
public UrlAliasManualData Update(UrlAliasManualData data);
```

Visual Basic

```
Public Function Update(data As UrlAliasManualData) As UrlAliasManualData
```

Parameters

Parameters	Description
UrlAliasManualData data	UrlAliasManualData (see page 1599) data

Returns

Returns the record with the updated field and any calculated field.

UrlAliasManual.ValidateName Method

Validates the Url Alais.

C#

```
public string ValidateName(string aliasName);
```

Visual Basic

```
Public Function ValidateName(aliasName As string) As string
```

Parameters

Parameters	Description
string aliasName	

Returns

An empty string if it passes validation or the error message.

UrlAliasRegEx Class

This class allows you to work with the Regular Expressions within aliasing.

Class Hierarchy

C#

```
public class UrlAliasRegEx : Ektron.Cms.API.CmsApi;
```

Visual Basic

```
Public Class UrlAliasRegEx
Inherits Ektron.Cms.API.CmsApi
```

File

UrlAliasRegEx.cs

UrlAliasRegEx Methods

The methods of the UrlAliasRegEx class are listed here.

Public Methods

	Name	Description
	Add (see page 941)	Adds a new regular expression based on UrlAliasRegExData (see page 1604) data.
	ClearCache (see page 942)	Refreshes the cache.
	Delete (see page 942)	Deletes an regular expression record from the database.
	EvaluateExpression (see page 942)	Evaluate an expression by passing the URL, expression and expression map.
	GetItem (see page 942)	Gets a single expression for a specified ID.
	GetList (see page 943)	Gets a list of either the active regular expression or a list of all expressions.
	GetPriorityList (see page 943)	Gets a list of page names for regular expressions alias types.
	GetTarget (see page 944)	Gets the target URL for a matching expression by providing the RegEx alias and server host name.
	Update (see page 944)	Updates an existing regular expression record based on UrlAliasRegExData (see page 1604) data.

UrlAliasRegEx.Add Method

Adds a new regular expression based on UrlAliasRegExData (see page 1604) data.

C#

```
public UrlAliasRegExData Add(UrlAliasRegExData data);
```

Visual Basic

```
Public Function Add(data As UrlAliasRegExData) As UrlAliasRegExData
```

Parameters

Parameters	Description
UrlAliasRegExData data	UrlAliasRegExData (see page 1604) data.

UrlAliasRegEx.ClearCache Method

Refreshes the cache.

C#

```
public void ClearCache();
```

Visual Basic

```
Public Function ClearCache() As void
```

UrlAliasRegEx.Delete Method

Deletes an regular expression record from the database.

C#

```
public void Delete(long regExId);
```

Visual Basic

```
Public Function Delete(regExId As long) As void
```

Parameters

Parameters	Description
long regExId	The ID of the regular expression.

UrlAliasRegEx.EvaluateExpression Method

Evaluate an expression by passing the URL, expression and expression map.

C#

```
public string EvaluateExpression(Uri url, string expression, string expressionMap);
```

Visual Basic

```
Public Function EvaluateExpression(url As Uri, expression As string, expressionMap As string)
As string
```

Parameters

Parameters	Description
Uri url	The URL to be evaluated.
string expression	The regular expression.
string expressionMap	The destination URL for the expression.

Returns

Returns the destination URL or an empty string if no match is found.

UrlAliasRegEx.GetItem Method

Gets a single expression for a specified ID.

C#

```
public UrlAliasRegExData GetItem(long regExId);
```

Visual Basic

```
Public Function GetItem(regExId As long) As UrlAliasRegExData
```

Parameters

Parameters	Description
long regExId	The ID of the regular expression.

Returns

A single expression for a specified ID.

UrlAliasRegEx.GetList Method

Gets a list of either the active regular expression or a list of all expressions.

C#

```
public List<UrlAliasRegExData> GetList(PagingInfo paging, long SiteId, bool onlyActive,
EkEnumeration.RegExAliasSearchField searchField, string searchValue,
EkEnumeration.RegExOrderBy orderBy);
```

Visual Basic

```
Public Function GetList(paging As PagingInfo, SiteId As long, onlyActive As bool, searchField
As EkEnumeration.RegExAliasSearchField, searchValue As string, orderBy As
EkEnumeration.RegExOrderBy) As List<UrlAliasRegExData>
```

Parameters

Parameters	Description
PagingInfo paging	The paging object.
bool onlyActive	Set to true to return only the active RegEx.
EkEnumeration.RegExAliasSearchField searchField	Decide which field is being searched. Choices are: • None • All • ExpressionName • TransformedUrl
string searchValue	The value for which to search.
EkEnumeration.RegExOrderBy orderBy	Select how to order the results. Choices are: • None • Priority • Expression • Target • Active • ExpressionName • TransformedUrl

Returns

Returns either the active regular expression or a list of all expressions.

UrlAliasRegEx.GetPriorityList Method

Gets a list of page names for regular expressions alias types.

C#

```
public List<EkEnumeration.RegExPriority> GetPriorityList();
```

Visual Basic

```
Public Function GetPriorityList() As List<EkEnumeration.RegExPriority>
```

Returns

A list of page names for regular expressions alias types.

UrlAliasRegEx.GetTarget Method

Gets the target URL for a matching expression by providing the RegEx alias and server host name.

C#

```
public string GetTarget(string urlAlias, string host);
```

Visual Basic

```
Public Function GetTarget(urlAlias As string, host As string) As string
```

Parameters

Parameters	Description
string host	The server's host name. For example, if your site is www.example.com, enter **example**.
url	The RegEx aliased URL.

Returns

The target URL for a matching expression.

UrlAliasRegEx.Update Method

Updates an existing regular expression record based on UrlAliasRegExData (see page 1604) data.

C#

```
public UrlAliasRegExData Update(UrlAliasRegExData data);
```

Visual Basic

```
Public Function Update(data As UrlAliasRegExData) As UrlAliasRegExData
```

Parameters

Parameters	Description
UrlAliasRegExData data	UrlAliasRegExData (see page 1604) data

UrlAliasSettings Class

This class allows you to work with settings for enabling and disabling the various types of aliasing available in CMS400.NET. In addition, you can work with alias settings for caching, tracing and page extensions. These settings reflect the same settings in the **Workarea** > **Settings** > **Configuration** > **URL Aliasing** > **Settings** screen.

Class Hierarchy

C#

```
public class UrlAliasSettings : Ektron.Cms.API.CmsApi;
```

Visual Basic

```
Public Class UrlAliasSettings
Inherits Ektron.Cms.API.CmsApi
```

File

UrlAliasSettings.cs

UrlAliasSettings Methods

The methods of the UrlAliasSettings class are listed here.

Public Methods

	Name	Description
	RefreshSettingsCache (see page 945)	Refreshes the settings cache.

UrlAliasSettings.RefreshSettingsCache Method

Refreshes the settings cache.

C#

```
public void RefreshSettingsCache();
```

Visual Basic

```
Public Function RefreshSettingsCache() As void
```

UrlAliasSettings Properties

The properties of the UrlAliasSettings class are listed here.

Public Properties

	Name	Description
	AutoCacheDuration (see page 946)	The time for which the page should reside in cache in seconds.
	AutoCacheSize (see page 946)	The maximum amount of items to store in the cache. If EnableAutoCache is set to true, this number reflects the number of automatic aliases that your server will cache. If this number is increased, it is possible that your performance will improve. But at some point, the size of the cache may degrade your server's performance.
	DefaultPage (see page 947)	Get or set the default page for the site. This page appears if someone enters your Web site's address without a page name. You can also specify a default document in IIS.
	DisableLanguageAwareness (see page 947)	This Setting will use the default language when it does not find an alis for the requested language.
	EnableAutoAliasing (see page 947)	Gets or sets whether automatic aliasing is enabled. • **True** = automatic aliasing is enabled. • **False** = automatic aliasing is disable.
	EnableAutoCaching (see page 947)	Enable or disable automatic aliasing caching. Enabling the automatic cache improves the server's performance. It minimizes the number of database by placing all automatic aliasing information in your server's memory. • **True** - enable automatic aliasing caching. • **False** - disable automatic aliasing caching.
	EnabledTracing (see page 947)	Enable or disable tracing. Tracing allows you to troubleshoot problems with aliasing. Developers can use this field to test Ektron's APIs for getting the target/destination URL. • **True** - enable tracing. • **False** - disable tracing.
	EnableManualAliasing (see page 948)	Gets or sets whether manual aliasing is enabled. • **True** = manual aliasing is enabled. • **False** = manual aliasing is disable.

	EnableManualCaching (see page 948)	Enable or disable the Manual aliasing cache. Enabling the manual cache improves the server's performance. It minimizes the number of database by placing all manual aliasing information in your server's memory. When you are developing a site, you might want to set this to false to make sure that everything is coming from the database. • **True** - enable manual aliasing cache. • **False** - disable manual aliasing cache.
	EnableRegExAliasing (see page 948)	Gets or sets whether RegEx aliasing is enabled. • **True** = RegEx aliasing is enabled. • **False** = RegEx aliasing is disable.
	EnableRegExCaching (see page 948)	Enable or disable the regular expression aliasing cache. • **True** - enable regular expression aliasing cache. • **False** - disable regular expression aliasing cache.
	IsAutoAliasEnabled (see page 949)	Enable or disable automatic aliasing. • **True** - enable automatic aliasing. • **False** - disable automatic aliasing.
	IsManualAliasEnabled (see page 949)	Enable or disable the manual aliasing. • **True** - enable manual aliasing. • **False** - disable manual aliasing.
	IsRegExAliasEnabled (see page 949)	Enable or disable regular expression aliasing. • **True** - enable regular expression aliasing. • **False** - disable regular expression aliasing.
	ManualCacheDuration (see page 949)	The time for which the page should reside in cache in seconds.
	ManualCacheSize (see page 949)	But, if you specify one here, the response time to display the page will be much quicker. The maximum amount of items to store in the cache.
	OverrideTemplate (see page 950)	The alias takes precedence over a templcate with the same name.
	PageExtensions (see page 950)	Get or set the page extension for which aliases can be created.
	QueryStringAction (see page 950)	The action to take when an url alias has a querystring.
	RegExCacheDuration (see page 950)	The time for which the page should reside in cache in seconds.
	RegExCacheSize (see page 950)	The maximum amount of items to store in the cache.

UrlAliasSettings.AutoCacheDuration Property

The time for which the page should reside in cache in seconds.

C#

```
public int AutoCacheDuration;
```

Visual Basic

```
Public Property AutoCacheDuration() As Integer
```

UrlAliasSettings.AutoCacheSize Property

The maximum amount of items to store in the cache. If EnableAutoCache is set to true, this number reflects the number of automatic aliases that your server will cache. If this number is increased, it is possible that your performance will improve. But at some point, the size of the cache may degrade your server's performance.

C#

```
public int AutoCacheSize;
```

Visual Basic

```
Public Property AutoCacheSize() As Integer
```

UrlAliasSettings.DefaultPage Property

Get or set the default page for the site. This page appears if someone enters your Web site's address without a page name. You can also specify a default document in IIS.

C#

```
public string DefaultPage;
```

Visual Basic

```
Public Property DefaultPage() As string
```

UrlAliasSettings.DisableLanguageAwareness Property

This Setting will use the default language when it does not find an alis for the requested language.

C#

```
public bool DisableLanguageAwareness;
```

Visual Basic

```
Public Property DisableLanguageAwareness() As bool
```

UrlAliasSettings.EnableAutoAliasing Property

Gets or sets whether automatic aliasing is enabled.

- **True** = automatic aliasing is enabled.
- **False** = automatic aliasing is disable.

C#

```
public bool EnableAutoAliasing;
```

Visual Basic

```
Public Property EnableAutoAliasing() As bool
```

UrlAliasSettings.EnableAutoCaching Property

Enable or disable automatic aliasing caching. Enabling the automatic cache improves the server's performance. It minimizes the number of database by placing all automatic aliasing information in your server's memory.

- **True** - enable automatic aliasing caching.
- **False** - disable automatic aliasing caching.

C#

```
public bool EnableAutoCaching;
```

Visual Basic

```
Public Property EnableAutoCaching() As bool
```

UrlAliasSettings.EnabledTracing Property

Enable or disable tracing. Tracing allows you to troubleshoot problems with aliasing. Developers can use this field to test Ektron's APIs for getting the target/destination URL.

- **True** - enable tracing.
- **False** - disable tracing.

C#

```
public bool EnabledTracing;
```

Visual Basic

```
Public Property EnabledTracing() As bool
```

UrlAliasSettings.EnableManualAliasing Property

Gets or sets whether manual aliasing is enabled.

- **True** = manual aliasing is enabled.
- **False** = manual aliasing is disable.

C#

```
public bool EnableManualAliasing;
```

Visual Basic

```
Public Property EnableManualAliasing() As bool
```

UrlAliasSettings.EnableManualCaching Property

Enable or disable the Manual aliasing cache. Enabling the manual cache improves the server's performance. It minimizes the number of database by placing all manual aliasing information in your server's memory. When you are developing a site, you might want to set this to false to make sure that everything is coming from the database.

- **True** - enable manual aliasing cache.
- **False** - disable manual aliasing cache.

C#

```
public bool EnableManualCaching;
```

Visual Basic

```
Public Property EnableManualCaching() As bool
```

UrlAliasSettings.EnableRegExAliasing Property

Gets or sets whether RegEx aliasing is enabled.

- **True** = RegEx aliasing is enabled.
- **False** = RegEx aliasing is disable.

C#

```
public bool EnableRegExAliasing;
```

Visual Basic

```
Public Property EnableRegExAliasing() As bool
```

UrlAliasSettings.EnableRegExCaching Property

Enable or disable the regular expression aliasing cache.

- **True** - enable regular expression aliasing cache.
- **False** - disable regular expression aliasing cache.

C#

```
public bool EnableRegExCaching;
```

Visual Basic

```
Public Property EnableRegExCaching() As bool
```

UrlAliasSettings.IsAutoAliasEnabled Property

Enable or disable automatic aliasing.

- **True** - enable automatic aliasing.
- **False** - disable automatic aliasing.

C#

```
public bool IsAutoAliasEnabled;
```

Visual Basic

```
Public Property IsAutoAliasEnabled() As bool
```

UrlAliasSettings.IsManualAliasEnabled Property

Enable or disable the manual aliasing.

- **True** - enable manual aliasing.
- **False** - disable manual aliasing.

C#

```
public bool IsManualAliasEnabled;
```

Visual Basic

```
Public Property IsManualAliasEnabled() As bool
```

UrlAliasSettings.IsRegExAliasEnabled Property

Enable or disable regular expression aliasing.

- **True** - enable regular expression aliasing.
- **False** - disable regular expression aliasing.

C#

```
public bool IsRegExAliasEnabled;
```

Visual Basic

```
Public Property IsRegExAliasEnabled() As bool
```

UrlAliasSettings.ManualCacheDuration Property

The time for which the page should reside in cache in seconds.

C#

```
public int ManualCacheDuration;
```

Visual Basic

```
Public Property ManualCacheDuration() As Integer
```

UrlAliasSettings.ManualCacheSize Property

But, if you specify one here, the response time to display the page will be much quicker. The maximum amount of items to store in the cache.

C#

```
public int ManualCacheSize;
```

Visual Basic

```
Public Property ManualCacheSize() As Integer
```

UrlAliasSettings.OverrideTemplate Property

The alias takes precedence over a templcate with the same name.

C#

```
public bool OverrideTemplate;
```

Visual Basic

```
Public Property OverrideTemplate() As bool
```

UrlAliasSettings.PageExtensions Property

Get or set the page extension for which aliases can be created.

C#

```
public string PageExtensions;
```

Visual Basic

```
Public Property PageExtensions() As string
```

UrlAliasSettings.QueryStringAction Property

The action to take when an url alias has a querystring.

C#

```
public EkEnumeration.QueryStringActionType QueryStringAction;
```

Visual Basic

```
Public Property QueryStringAction() As EkEnumeration.QueryStringActionType
```

UrlAliasSettings.RegExCacheDuration Property

The time for which the page should reside in cache in seconds.

C#

```
public int RegExCacheDuration;
```

Visual Basic

```
Public Property RegExCacheDuration() As Integer
```

UrlAliasSettings.RegExCacheSize Property

The maximum amount of items to store in the cache.

C#

```
public int RegExCacheSize;
```

Visual Basic

```
Public Property RegExCacheSize() As Integer
```

Ektron.Cms.API.User Namespace

This is namespace Ektron.Cms.API.User.

Classes

	Name	Description
	ADUser (see page 951)	This is class Ektron.Cms.API.User.ADUser.
	User (see page 958)	This is class Ektron.Cms.API.User.User.

Classes

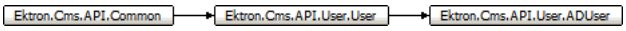
The following table lists classes in this documentation.

Classes

	Name	Description
	ADUser (see page 951)	This is class Ektron.Cms.API.User.ADUser.
	User (see page 958)	This is class Ektron.Cms.API.User.User.

ADUser Class

Class Hierarchy



C#

```
public class ADUser : User;
```

Visual Basic

```
Public Class ADUser
Inherits User
```

File

ADUser.vb

Description

This is class Ektron.Cms.API.User.ADUser.

ADUser Methods

The methods of the ADUser class are listed here.

Public Methods

	Name	Description
	AddADGroupToCMS (see page 952)	This function allows you add Active Directory (see page 196) groups to the CMS.
	AddADUsersToCMSByUsername (see page 952)	Adds Active Directory (see page 196) users to the CMS by their user name and domain.
	DeSynchUserGroups (see page 952)	Updates the group with a new group name.
	DeSynchUsers (see page 953)	Update the user with new user details.
	GetADMapping (see page 953)	Loads the CMS and Active Directory (see page 196) name.
	GetADStatus (see page 953)	Loads the required Synch, Desynch and relationship flags of the configured Active Directory (see page 196).
	GetAvailableADGroups (see page 954)	Loads all of the Active Directory (see page 196) groups for a given domain and filter.
	GetAvailableADUsers (see page 954)	Loads all available Active Directory (see page 196) users.
	GetCMSGroupsToSync (see page 954)	Loads all the CMS groups that need to be synchronized.
	GetCMSRelationshipsToSync (see page 955)	Loads all the CMS relationships that need to be synchronized with Active Directory (see page 196).

	GetCMSUsersToSync (see page 955)	Loads all the CMS users that need to be synchronized with Active Directory (see page 196).
	GetDomains (see page 955)	Loads a list of domains from Active Directory (see page 196).
	MapCMSUserGroupToAD (see page 956)	Updates the given CMS group corresponding to the Active Directory (see page 196) group.
	MapCMSUserToAD (see page 956)	Maps the given user ID to the corresponding Active Directory (see page 196) entry.
	SynchCMSGroupsToAD (see page 956)	Synchronizes the CMS group data to the Active Directory (see page 196) group data. This updates the CMS data with the corresponding to Active Directory (see page 196) changes.
	SynchCMSRelationshipsToAD (see page 957)	Synchronizes the CMS relations, such as groups, to the Active Directory (see page 196) for the given users.
	SynchCMSUsersToAD (see page 957)	Synchronize the CMS user data to the Active Directory (see page 196) data.
	UpdateADMapping (see page 957)	Updates the Active directory information into the CMS based on the user ID and user data.
	UpdateADUsersGroups (see page 958)	Updates Active Directory (see page 196) users in a CMS group.
	UpdateUsersInfoFromAD (see page 958)	Updates the user details from the Active Directory (see page 196) details.

ADUser.AddADGroupToCMS Method

This function allows you add Active Directory (see page 196) groups to the CMS.

C#

```
public AddADGroupToCMS(Object GroupPaths);
```

Visual Basic

```
Public Sub AddADGroupToCMS(ByVal GroupPaths As Object)
```

Parameters

Parameters	Description
ByVal GroupPaths As Object	A collection of Active Directory (see page 196) group paths.

ADUser.AddADUsersToCMSByUsername Method

Adds Active Directory (see page 196) users to the CMS by their user name and domain.

C#

```
public AddADUsersToCMSByUsername(Collection UserNames, Collection UserDomains);
```

Visual Basic

```
Public Sub AddADUsersToCMSByUsername(ByVal UserNames As Collection, ByVal UserDomains As
Collection)
```

Parameters

Parameters	Description
ByVal UserNames As Collection	A collection of user names.
ByVal UserDomains As Collection	A collection of domain names.

ADUser.DeSynchUserGroups Method

Updates the group with a new group name.

C#

```
public AdDeSyncGroupData[] DeSynchUserGroups(Boolean Flag);
```

Visual Basic

```
Public Function DeSynchUserGroups(ByVal Flag As Boolean) As AdDeSyncGroupData()
```

Parameters

Parameters	Description
ByVal Flag As Boolean	If set to true, updates the user group with the new group information. Otherwise, nothing is updated.

Returns

Ektron.Cms.AdDeSyncGroupData()

ADUser.DeSynchUsers Method

Update the user with new user details.

C#

```
public AdDeSyncUserData[] DeSynchUsers(Boolean Flag);
```

Visual Basic

```
Public Function DeSynchUsers(ByVal Flag As Boolean) As AdDeSyncUserData()
```

Parameters

Parameters	Description
ByVal Flag As Boolean	If set to true, updates the user with new user information. Otherwise, nothing is updated.

Returns

Ektron.Cms.AdDeSyncUserData()

ADUser.GetADMapping Method

Loads the CMS and Active Directory (see page 196) name.

C#

```
public Overloads AdMappingData[] GetADMapping(Long CallerId, String MapType, int Map, int
Exclude, int Editable);
```

Visual Basic

```
Public Overloads Function GetADMapping(ByVal CallerId As Long, ByVal MapType As String, ByVal
Map As Integer, ByVal Exclude As Integer, ByVal Editable As Integer) As AdMappingData()
```

Parameters

Parameters	Description
ByVal CallerId As Long	The ID of the who is calling this function.
ByVal MapType As String	Possible value is “userprop” (user properties).
ByVal Map As Integer	Ektron.Cms.Common.EkEnumeration.MapTypes (include, exclude and dontconsider are the possible values).
ByVal Exclude As Integer	Flag (see page 1231) (0 – false, 1- true).
ByVal Editable As Integer	Flag (see page 1231) (0 - false, 1 - true).

Returns

Ektron.Cms.AdMappingData()

ADUser.GetADStatus Method

Loads the required Synch, Desynch and relationship flags of the configured Active Directory (see page 196).

C#

```
public AdSyncData GetADStatus();
```

Visual Basic

```
Public Function GetADStatus() As AdSyncData
```

Returns

Ektron.Cms.AdSyncData

ADUser.GetAvailableADGroups Method

Loads all of the Active Directory (see page 196) groups for a given domain and filter.

C#

```
public GroupData[] GetAvailableADGroups(String Filter, String Domain);
```

Visual Basic

```
Public Function GetAvailableADGroups(ByVal Filter As String, ByVal Domain As String) As
GroupData()
```

Parameters

Parameters	Description
ByVal Filter As String	Filter by complete or a specific group name.
ByVal Domain As String	The domain name. (Required)

Returns

Ektron.Cms.GroupData()

ADUser.GetAvailableADUsers Method

Loads all available Active Directory (see page 196) users.

C#

```
public UserData[] GetAvailableADUsers(Object Attributes, Object Filter, String OrderBy, String
Domain);
```

Visual Basic

```
Public Function GetAvailableADUsers(ByVal Attributes As Object, ByVal Filter As Object, ByVal
OrderBy As String, ByVal Domain As String) As UserData()
```

Parameters

Parameters	Description
ByVal Attributes As Object	Possible attributes are firstname, lastname, username or any(*).
ByVal Filter As Object	Possible filters are firstname, lastname, username or any(*).
ByVal OrderBy As String	Any specific order such as username, firstname, lastname and domain.
ByVal Domain As String	The active directory domain. (Required)

Returns

Ektron.Cms.UserData()

ADUser.GetCMSGroupsToSync Method

Loads all the CMS groups that need to be synchronized.

C#

```
public GroupData[] GetCMSGroupsToSync(int MaxReturn);
```

Visual Basic

```
Public Function GetCMSGroupsToSync(ByVal MaxReturn As Integer) As GroupData()
```

Parameters

Parameters	Description
ByVal MaxReturn As Integer	The maximum number of groups that should be displayed.

Returns

Ektron.Cms.GroupData()

ADUser.GetCMSRelationshipsToSync Method

Loads all the CMS relationships that need to be synchronized with Active Directory (see page 196).

C#

```
public UserGroupData[] GetCMSRelationshipsToSync(int MaxReturn);
```

Visual Basic

```
Public Function GetCMSRelationshipsToSync(ByVal MaxReturn As Integer) As UserGroupData()
```

Parameters

Parameters	Description
ByVal MaxReturn As Integer	The maximum number of results that should be displayed.

Returns

Ektron.Cms.UserGroupData()

ADUser.GetCMSUsersToSync Method

Loads all the CMS users that need to be synchronized with Active Directory (see page 196).

C#

```
public UserData[] GetCMSUsersToSync(int MaxReturn);
```

Visual Basic

```
Public Function GetCMSUsersToSync(ByVal MaxReturn As Integer) As UserData()
```

Parameters

Parameters	Description
ByVal MaxReturn As Integer	The maximum number of users that should be displayed.

Returns

Ektron.Cms.UserData()

ADUser.GetDomains Method

Loads a list of domains from Active Directory (see page 196).

C#

```
public DomainData[] GetDomains(int ShowAll, int CheckIntegration);
```

Visual Basic

```
Public Function GetDomains(ByVal ShowAll As Integer, ByVal CheckIntegration As Integer) As
DomainData()
```

Parameters

Parameters	Description
ByVal ShowAll As Integer	flag (1 = show all).

ByVal CheckIntegration As Integer	Check whether CMS is configured to integrate Active Directory (see page 196). Enter **1** to check integration. Otherwise, the method continues without checking.

Returns

Ektron.Cms.DomainData()

ADUser.MapCMSUserGroupToAD Method

Updates the given CMS group corresponding to the Active Directory (see page 196) group.

C#

```
public MapCMSUserGroupToAD(Long GroupId, String GroupName, String Domain);
```

Visual Basic

```
Public Sub MapCMSUserGroupToAD(ByVal GroupId As Long, ByVal GroupName As String, ByVal Domain
As String)
```

Parameters

Parameters	Description
ByVal GroupId As Long	The CMS group ID to update.
ByVal GroupName As String	The CMS group name to update.
ByVal Domain As String	The Active Directory (see page 196) domain.

ADUser.MapCMSUserToAD Method

Maps the given user ID to the corresponding Active Directory (see page 196) entry.

C#

```
public MapCMSUserToAD(Long UserId, String UserName, String Domain, int PullInfo);
```

Visual Basic

```
Public Sub MapCMSUserToAD(ByVal UserId As Long, ByVal UserName As String, ByVal Domain As
String, ByVal PullInfo As Integer)
```

Parameters

Parameters	Description
ByVal UserId As Long	The user's ID to map.
ByVal UserName As String	The user's name to map.
ByVal Domain As String	The user's domain.
ByVal PullInfo As Integer	Flag (see page 1231) to load Active Directory (see page 196) information. (1 = Load Active Directory (see page 196) information, otherwise none.)

ADUser.SynchCMSGroupsToAD Method

Synchronizes the CMS group data to the Active Directory (see page 196) group data. This updates the CMS data with the corresponding to Active Directory (see page 196) changes.

C#

```
public SynchCMSGroupsToAD(Collection GroupIds, Collection UserNames, Collection Domains,
Collection Actions);
```

Visual Basic

```
Public Sub SynchCMSGroupsToAD(ByVal GroupIds As Collection, ByVal UserNames As Collection,
ByVal Domains As Collection, ByVal Actions As Collection)
```

Parameters

Parameters	Description
ByVal GroupIds As Collection	A collection of group IDs.

ByVal UserNames As Collection	A collection of usernames.
ByVal Domains As Collection	A collection of domains.
ByVal Actions As Collection	Possible actions are “delete” or “map”.

ADUser.SynchCMSRelationshipsToAD Method

Synchronizes the CMS relations, such as groups, to the Active Directory (see page 196) for the given users.

C#

```
public SynchCMSRelationshipsToAD(Collection UserIds, Collection GroupIds, Collection Actions);
```

Visual Basic

```
Public Sub SynchCMSRelationshipsToAD(ByVal UserIds As Collection, ByVal GroupIds As
Collection, ByVal Actions As Collection)
```

Parameters

Parameters	Description
ByVal UserIds As Collection	A collection of user IDs.
ByVal GroupIds As Collection	A collection of group IDs.
ByVal Actions As Collection	A collection of actions (possible action is “delete”).

ADUser.SynchCMSUsersToAD Method

Synchronize the CMS user data to the Active Directory (see page 196) data.

C#

```
public SynchCMSUsersToAD(Collection UserIds, Collection UserNames, Collection Domains,
Collection Actions, int PullInfo);
```

Visual Basic

```
Public Sub SynchCMSUsersToAD(ByVal UserIds As Collection, ByVal UserNames As Collection, ByVal
Domains As Collection, ByVal Actions As Collection, ByVal PullInfo As Integer)
```

Parameters

Parameters	Description
ByVal UserIds As Collection	A collection of user IDs.
ByVal UserNames As Collection	A collection of usernames.
ByVal Domains As Collection	A collection of domains.
ByVal Actions As Collection	A collection of actions corresponding to items added to the collection. The possible collection values are “delete” or “map”.
ByVal PullInfo As Integer	Retrieves the Active Directory (see page 196) information when the value is 1, otherwise none.

ADUser.UpdateADMapping Method

Updates the Active directory information into the CMS based on the user ID and user data.

C#

```
public UpdateADMapping(Long UserId, Object pagedata);
```

Visual Basic

```
Public Sub UpdateADMapping(ByVal UserId As Long, ByVal pagedata As Object)
```

Parameters

Parameters	Description
ByVal UserId As Long	The user’s ID.
ByVal pagedata As Object	Key, value pair for given user ID.

ADUser.UpdateADUsersGroups Method

Updates Active Directory (see page 196) users in a CMS group.

C#

```
public UpdateADUsersGroups(String UserName, String Domain);
```

Visual Basic

```
Public Sub UpdateADUsersGroups(ByVal UserName As String, ByVal Domain As String)
```

Parameters

Parameters	Description
ByVal UserName As String	The user's username.
ByVal Domain As String	The user's Domain.

ADUser.UpdateUsersInfoFromAD Method

Updates the user details from the Active Directory (see page 196) details.

C#

```
public UpdateUsersInfoFromAD(String UserName, String Domain);
```

Visual Basic

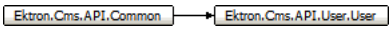
```
Public Sub UpdateUsersInfoFromAD(ByVal UserName As String, ByVal Domain As String)
```

Parameters

Parameters	Description
ByVal UserName As String	The user's username.
ByVal Domain As String	The user's Domain.

User Class

Class Hierarchy



C#

```
public class User : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class User
Inherits Ektron.Cms.API.Common
```

File

User.vb

Description

This is class Ektron.Cms.API.User.User.

User Methods

The methods of the User class are listed here.

Public Methods

	Name	Description
	ActivateUserAccount (see page 960)	This is the overview for the ActivateUserAccount method overload.

	AddMembershipUser (see page 961)	Adds a new membership user to CMS.
	AddNewUser (see page 963)	**This method is obsolete. Please use the AddMembershipUser (see page 961) method.** Adds a new membership user to CMS.
	AddRolePermission (see page 964)	Adds permission for the given role.
	AddUser (see page 964)	This is the overview for the AddUser method overload.
	AddUserGroup (see page 965)	Creates new user group.
	AddUserToGroup (see page 965)	Adds an existing user to an existing group.
	AutoLogInUser (see page 965)	Login (see page 347) the given user credentials.
	CreateBlankUserObject (see page 966)	This is the overview for the CreateBlankUserObject method overload.
	DeleteUserByID (see page 967)	Removes the user from the system. This will not purge the data, but it disables the user from further accessing the system.
	DeleteUserByIds (see page 967)	Removes the user from the system. This will not purge the data, but it disables the user from further accessing the system.
	DeleteUserFromGroup (see page 967)	Removes a user from the designated group.
	DeleteUserGroup (see page 967)	Removes the group from the system permanently.
	EditUserCustomProperties (see page 968)	This method creates the editable fields(UI) for user's custom properties.
	GetActiveUser (see page 968)	Returns all of the user properties with or without preference.
	GetADMapping (see page 968)	Loads the CMS and Active Directory (see page 196) name.
	GetAllCmsUsers (see page 969)	Loads all the CMS users.
	GetAllCustomProperties (see page 969)	Returns array of all the custom properties defined in the system.
	GetAllUserByCustomProperty (see page 970)	Returns all the users based on the given properties.
	GetAllUserGroups (see page 970)	This is the overview for the GetAllUserGroups method overload.
	GetAllUsers (see page 971)	This is the overview for the GetAllUsers method overload.
	GetGroupsUserIsIn (see page 972)	Loads all the groups by user.
	GetMembershipUser (see page 972)	Returns user data based on the user's login name (username).
	GetNonVerifyUser (see page 973)	This is the overview for the GetNonVerifyUser method overload.
	GetUser (see page 973)	Loads the user details for the given user.
	GetUserByUsername (see page 974)	Loads the user details for the given user.
	GetUserCustomProperties (see page 975)	Returns an array of custom properties for a given user's ID.
	GetUserEmailInfo (see page 976)	Returns user data for a given user's ID.
	GetUserForFolderAdmin (see page 976)	
	GetUserGroup (see page 976)	Loads a given group's details.
	GetUserGroupByName (see page 976)	Loads a given group's details.
	GetUserGroupForFolderAdmin (see page 977)	Returns all the group's data for the administrator.
	GetUserPermissions (see page 977)	Loads the user permissions as a array of UserPermissionData.
	GetUserPreference (see page 978)	Loads the preferences for the given user.

	GetUsers (see page 978)	This is the overview for the GetUsers method overload.
	GetUsersForTask (see page 979)	Loads the users that have permissions to the given content.
	GetUsersNotInGroup (see page 980)	Loads all the users that are not in the given group.
	IsAdmin (see page 980)	Determines the if the current user is an Administrator.
	IsAGroupMember (see page 980)	For a given userId, returns whether that user is a member of the group.
	IsARoleMember (see page 981)	This is the overview for the IsARoleMember method overload.
	IsARoleMemberForContent (see page 984)	Determines whether a Role Member is allowed to work with a given piece of content as an administrator or not. Administrations privileges are for the given piece of content only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.
	IsARoleMemberForContent_FolderUserAdmin (see page 984)	Determines whether a Role Member is allowed to work with a given conatent as an administrator or not. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.
	IsARoleMemberForFolder (see page 984)	Determines whether a Role Member is allowed to work with a given folder as an administrator or not. Administrations privileges are for the given folder only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.
	IsARoleMemberForFolder_FolderUserAdmin (see page 985)	Determines whether a Role Member is allowed to work with a given folder as an administrator or not. Administrations privileges are for the given folder only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.
	IsARoleMemberForFolder_MetadataAdmin (see page 985)	Determines whether a Role Member is allowed to work with a given folder as an meta data administrator or not.
	LockUserAccount (see page 986)	This function locks a user account.
	LogInUser (see page 986)	This is the overview for the LogInUser method overload.
	ReadCustomProperties (see page 990)	Returns the custom properties as a Visual Basic collection. Importing the Microsoft.VisualBasic name space might be required.
	ResetMembershipUserPassword (see page 990)	This is the overview for the ResetMembershipUserPassword method overload.
	SetAuthenticationCookie (see page 991)	This method sets the authentication cookie.
	UnsubscribeUser (see page 991)	Unsubscribes a user from the system.
	UpdateCustomPropertiesItemOrder (see page 992)	Updates the order of available custom properties.
	UpdateUser (see page 992)	This is the overview for the UpdateUser method overload.
	UpDateUserGroup (see page 993)	This is the overview for the UpDateUserGroup method overload.
	UpdateUserPreferences (see page 993)	Updates the given user's preference data.
	UserObject (see page 994)	This is the overview for the UserObject method overload.

ActivateUserAccount Method

This is the overview for the ActivateUserAccount method overload.

Description

Activates a deleted (deactivated) user into the current/active state using the UserId or the Username and GUID (AccountID) parameters.

This method is overloaded

Overload List

	Name	Description
	User.ActivateUserAccount (Long) (see page 961)	Activates a deleted (deactivated) user into the current/active state using the UserId. This function is overloaded. See Also: ActivateUserAccount (String String)
	User.ActivateUserAccount (String, String) (see page 961)	Activates the deleted (deactivated) user into the current/active state using the Username and GUID (AccountID) parameter. This function is overloaded. See Also: ActivateUserAccount (Long)

User.ActivateUserAccount Method (Long)

Activates a deleted (deactivated) user into the current/active state using the UserId.

This function is overloaded. See Also: ActivateUserAccount (String String)

C#

```
public Boolean ActivateUserAccount(Long UserId);
```

Visual Basic

```
Public Function ActivateUserAccount(ByVal UserId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal UserId As Long	A unique ID for the required user.

Returns

Boolean

User.ActivateUserAccount Method (String, String)

Activates the deleted (deactivated) user into the current/active state using the Username and GUID (AccountID) parameter.

This function is overloaded. See Also: ActivateUserAccount (Long)

C#

```
public Boolean ActivateUserAccount(String Username, String AccountID);
```

Visual Basic

```
Public Function ActivateUserAccount(ByVal Username As String, ByVal AccountID As String) As
Boolean
```

Parameters

Parameters	Description
ByVal Username As String	The login name for the user.
ByVal AccountID As String	The GUID for the user.

Returns

Boolean

User.AddMembershipUser Method

Adds a new membership user to CMS.

C#

```
public UserData AddMembershipUser(UserData udUserData);
```

Visual Basic

```
Public Function AddMembershipUser(ByVal udUserData As UserData) As UserData
```

Parameters

Parameters	Description
ByVal udUserData As UserData	An object of class which represents the corresponding user elements.

Returns

Ektron.Cms.UserData

Example

The following example shows how to create a Web page from which a new membership user can be added. This example uses some standard drag and drop controls and a small section of VB code utilizing the AddMembershipUser method. This method uses the InternalAdmin to add the Membership (see page 360) User (see page 958). In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, create a table with four rows and three columns.
2. In the first row, first column, add a label that shows where to enter the membership user's Email address.
```
<asp:Label ID="Label6" runat="server" Text="Email: " Width="76px"></asp:Label>
```
3. In the first row, second column, add a text box to enter the Email address.
```
<asp:TextBox ID="txtEmail" runat="server" Width="144px"></asp:TextBox>
```
4. In the first row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the Email address box.
```
<asp:RequiredFieldValidator ID="rfvEmail" runat="server" ErrorMessage="Please enter an
Email address"
      ControlToValidate="txtEmail" Display="Dynamic">Please enter email
address</asp:RequiredFieldValidator>
```
5. In the same table cell, add a RegularExpressionValidator control. This validates the information entered is in the format of an Email address.
```
<asp:RegularExpressionValidator ID="rfvEmailFormat" ControlToValidate="txtEmail"
runat="server" ErrorMessage="Please enter a valid email address"
     ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*" Width="236px"
></asp:RegularExpressionValidator>
```
6. In the second row, first column, add a label to show where to enter a first name.
```
<asp:Label ID="Label4" runat="server" Text="First name: " Width="76px"></asp:Label>
```
7. In the second row, second column, add a text box to enter the fist name.
```
<asp:TextBox ID="txtFirstName" runat="server" Width="144px"></asp:TextBox>
```
8. In the second row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the first name box.
```
<asp:RequiredFieldValidator ID="rfvFirstName" runat="server" ErrorMessage="Please enter
first name"
      ControlToValidate="txtFirstName">Please enter first name</asp:RequiredFieldValidator>
```
9. In the third row, first column, add a label to show where to enter a last name.
```
<asp:Label ID="Label7" runat="server" Text="Last Name:" Width="76px"></asp:Label>
```
10. In the third row, second column, add a text box to enter the last name.
```
<asp:TextBox ID="txtLastName" runat="server" Width="144px"></asp:TextBox>
```
11. In the third row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the last name box.
```
<asp:RequiredFieldValidator ID="rfvLastName" runat="server" ErrorMessage="Please enter last
name"
      ControlToValidate="txtLastName">Please enter last name</asp:RequiredFieldValidator>
```
12. In the fourth row, first column, add a label to show where to enter a password.
```
<asp:Label ID="Label8" runat="server" Text="Password: " Width="76px"></asp:Label>
```
13. In the fourth row, second column, add a text box to enter the password.
```
<asp:TextBox ID="txtPassword" runat="server" TextMode="Password"
Width="144px"></asp:TextBox>
```
14. In the fourth row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the password box.
```
<asp:RequiredFieldValidator ID="rfvPassword" runat="server" ErrorMessage="Please enter
password"
       ControlToValidate="txtPassword">Please enter password</asp:RequiredFieldValidator>
```
15. After the table, add a break and then add a button to submit the membership user's information.
```
<br />
<asp:Button ID="btnAdd" runat="server" Text="Add Membership User" Width="153px" />
```
16. Add two breaks for spacing purposes.
```
<br /><br />
```
17. Add a results label to display a message stating the membership user was added and the new ID.
```
<asp:Label ID="lblResult" runat="server"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add button click event.

```
        Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
btnAdd.Click
```

2. Create an object for the user data items.

```
            Dim memDetails As New Ektron.Cms.UserData
```

3. Create an object for the new membership user information.

```
            Dim newMemUser As New Ektron.Cms.UserData
```

4. Create an object for the User (see page 958) API (see page 669).

```
            Dim userAPI As New Ektron.Cms.API.User.User
```

5. Create an integer place holder for the new ID.

```
            Dim newMemUserId As Integer
```

6. Populate memDetails with the information from the text boxes.

```
            memDetails.Username = txtEmail.Text
            memDetails.FirstName = txtFirstName.Text
            memDetails.LastName = txtLastName.Text
            memDetails.Password = txtPassword.Text
```

7. Create a Try/Catch around the AddMembershipUser method. This allows you to catch any exceptions and helps when debugging.

```
            Try
                newMemUser = userAPI.AddMembershipUser(memDetails)
```

8. Assign newMemUserId a new ID.

```
                newMemUserId = newMemUser.Id
```

9. Check to see if there is an ID returned. If so, display a message with the membership username and ID.

```
            If (newMemUserId > 0) Then
                lblResult.Text = "The membership user " & txtEmail.Text & " was added into
CMS400.Net with UserID = " & newMemUserId
            End If
```

10. If there is an error, let the user know what happened.

```
            Catch ex As Exception
               lblResult.Text = ex.Message
            End Try

        End Sub
```

User.AddNewUser Method

This method is obsolete. Please use the AddMembershipUser (see page 961) method.

Adds a new membership user to CMS.

C#

```
[Obsolete("Use ""AddMembershipUser"" instead.")]
public UserData AddNewUser(UserData udUserData);
```

Visual Basic

```
<Obsolete("Use ""AddMembershipUser"" instead.")>
Public Function AddNewUser(ByVal udUserData As UserData) As UserData
```

Parameters

Parameters	Description
ByVal udUserData As UserData	An object of class which represents the corresponding user elements.

Returns

Ektron.Cms.UserData

User.AddRolePermission Method

Adds permission for the given role.

C#

```
public Boolean AddRolePermission(String RoleName);
```

Visual Basic

```
Public Function AddRolePermission(ByVal RoleName As String) As Boolean
```

Parameters

Parameters	Description
ByVal RoleName As String	Required role name.

Returns

True if successfully added. Otherwise, returns false

AddUser Method

This is the overview for the AddUser method overload.

Overload List

	Name	Description
	User.AddUser (Collection) (see page 964)	Adds a new user to CMS.
	User.AddUser (UserData) (see page 964)	Adds a new user to CMS.

User.AddUser Method (Collection)

Adds a new user to CMS.

C#

```
public UserData AddUser(Collection Users);
```

Visual Basic

```
Public Function AddUser(ByVal Users As Collection) As UserData
```

Parameters

Parameters	Description
ByVal Users As Collection	User (see page 958) details as a value, key pair.

Returns

Ektron.Cms.UserData

User.AddUser Method (UserData)

Adds a new user to CMS.

C#

```
public UserData AddUser(UserData udUserData);
```

Visual Basic

```
Public Function AddUser(ByVal udUserData As UserData) As UserData
```

Parameters

Parameters	Description
ByVal udUserData As UserData	User (see page 958) details as a value strongly typed UserData object.

Returns

Ektron.Cms.UserData

User.AddUserGroup Method

Creates new user group.

C#

```
public Collection AddUserGroup(String GroupName, Optional ByVal String GroupPath, Optional
ByVal String GroupDomain);
```

Visual Basic

```
Public Function AddUserGroup(ByVal GroupName As String, Optional ByVal GroupPath As String =
"", Optional ByVal GroupDomain As String = "") As Collection
```

Parameters

Parameters	Description
ByVal GroupName As String	The name of the group.
Optional ByVal GroupPath As String = ""	Group path, usually required for Active Directory (see page 196), otherwise CMS group assumed.
Optional ByVal GroupDomain As String = ""	Group domain, usually required for Active Directory (see page 196).

User.AddUserToGroup Method

Adds an existing user to an existing group.

C#

```
public AddUserToGroup(Long UserId, Long GroupId);
```

Visual Basic

```
Public Sub AddUserToGroup(ByVal UserId As Long, ByVal GroupId As Long)
```

Parameters

Parameters	Description
ByVal UserId As Long	The user's ID.
ByVal GroupId As Long	The required group's ID.

User.AutoLogInUser Method

Login (see page 347) the given user credentials.

C#

```
public UserData AutoLogInUser(String SAMusername, String domain, String ServerName);
```

Visual Basic

```
Public Function AutoLogInUser(ByVal SAMusername As String, ByVal domain As String, ByVal
ServerName As String) As UserData
```

Parameters

Parameters	Description
ByVal SAMusername As String	user name
ByVal domain As String	domain
ByVal ServerName As String	server name

Returns

True if the user successfully logged in otherwise false.

CreateBlankUserObject Method

This is the overview for the CreateBlankUserObject method overload.

Description

This method is overloaded.

Overload List

	Name	Description
	User.CreateBlankUserObject () (see page 966)	Returns the instance of the user data. This function is overloaded. See Also: CreateBlankUserObject (UserData)
	User.CreateBlankUserObject (UserData) (see page 966)	Sets the given user data values into the UserData class. This function is overloaded. See Also: CreateBlankUserObject

User.CreateBlankUserObject Method ()

Returns the instance of the user data.

This function is overloaded. See Also: CreateBlankUserObject (UserData)

C#

```
public UserData CreateBlankUserObject();
```

Visual Basic

```
Public Function CreateBlankUserObject() As UserData
```

Returns

Ektron.Cms.UserData

User.CreateBlankUserObject Method (UserData)

Sets the given user data values into the UserData class.

This function is overloaded. See Also: CreateBlankUserObject (see page 966)

C#

```
public CreateBlankUserObject(ref UserData udUserData);
```

Visual Basic

```
Public Sub CreateBlankUserObject(ByRef udUserData As UserData)
```

Parameters

Parameters	Description
ByRef udUserData As UserData	An Object of user properties.

Remarks

User.DeleteUserByID Method

Removes the user from the system. This will not purge the data, but it disables the user from further accessing the system.

C#

```
public DeleteUserByID(Long Id);
```

Visual Basic

```
Public Sub DeleteUserByID(ByVal Id As Long)
```

Parameters

Parameters	Description
ByVal Id As Long	The user's ID.

User.DeleteUserByIds Method

Removes the user from the system. This will not purge the data, but it disables the user from further accessing the system.

C#

```
public DeleteUserByIds(String Ids);
```

Visual Basic

```
Public Sub DeleteUserByIds(ByVal Ids As String)
```

Parameters

Parameters	Description
ByVal Ids As String	The selected user IDs (comma separated values).

User.DeleteUserFromGroup Method

Removes a user from the designated group.

C#

```
public DeleteUserFromGroup(Long UserId, Long GroupId, Optional ByVal Boolean ReturnDeleted);
```

Visual Basic

```
Public Sub DeleteUserFromGroup(ByVal UserId As Long, ByVal GroupId As Long, Optional ByVal
ReturnDeleted As Boolean = False)
```

Parameters

Parameters	Description
ByVal UserId As Long	The user's ID.
ByVal GroupId As Long	The group's ID.

User.DeleteUserGroup Method

Removes the group from the system permanently.

C#

```
public DeleteUserGroup(Long Id);
```

Visual Basic

```
Public Sub DeleteUserGroup(ByVal Id As Long)
```

Parameters

Parameters	Description
ByVal Id As Long	The group's ID.

User.EditUserCustomProperties Method

This method creates the editable fields(UI) for user's custom properties.

C#

```
public String EditUserCustomProperties(Optional ByVal Long UserId, Optional ByVal Boolean
bReadOnly);
```

Visual Basic

```
Public Function EditUserCustomProperties(Optional ByVal UserId As Long = 0, Optional ByVal
bReadOnly As Boolean = False) As String
```

Parameters

Parameters	Description
Optional ByVal UserId As Long = 0	User (see page 958)'s ID. If the user ID is not specified then it will return empty fields.
Optional ByVal bReadOnly As Boolean = False	

Returns

String

User.GetActiveUser Method

Returns all of the user properties with or without preference.

C#

```
public UserData GetActiveUser(Long Id, Optional ByVal Boolean Preference);
```

Visual Basic

```
Public Function GetActiveUser(ByVal Id As Long, Optional ByVal Preference As Boolean = False)
As UserData
```

Parameters

Parameters	Description
ByVal Id As Long	The user's ID.
Optional ByVal Preference As Boolean = False	Preference is returned only if the value true, otherwise nothing is returned for preference.

Returns

Ektron.Cms.UserData

User.GetADMapping Method

Loads the CMS and Active Directory (see page 196) name.

C#

```
public AdMappingData[] GetADMapping(Long CallerId, String MapType, int Map, int Exclude, int
Editable);
```

Visual Basic

```
Public Function GetADMapping(ByVal CallerId As Long, ByVal MapType As String, ByVal Map As
Integer, ByVal Exclude As Integer, ByVal Editable As Integer) As AdMappingData()
```

Parameters

Parameters	Description
ByVal CallerId As Long	Who is calling this function.
ByVal MapType As String	Possible value is “userprop” (user properties).
ByVal Map As Integer	Ektron.Cms.Common.EkEnumeration.MapTypes (include, exclude and dontconsider are the possible values).
ByVal Exclude As Integer	Flag (see page 1231) (0 – false, 1-true).
ByVal Editable As Integer	Flag (see page 1231) (0-false, 1-true).

Returns

Ektron.Cms.AdMappingData()

User.GetAllCmsUsers Method

Loads all the CMS users.

C#

```
public UserData[] GetAllCmsUsers(String OrderBy);
```

Visual Basic

```
Public Function GetAllCmsUsers(ByVal OrderBy As String) As UserData()
```

Parameters

Parameters	Description
ByVal OrderBy As String	Used to arrange data in a specific order. The string value can be: • username • firstname • lastname • userid • language • datecreated • lastlogindate

Returns

Ektron.Cms.UserData()

Description

User.GetAllCustomProperties Method

Returns array of all the custom properties defined in the system.

C#

```
public CustomAttribute[] GetAllCustomProperties();
```

Visual Basic

```
Public Function GetAllCustomProperties() As CustomAttribute()
```

Returns

Ektron.Cms.CustomAttribute()

User.GetAllUserByCustomProperty Method

Returns all the users based on the given properties.

C#

```
public UserData[]
GetAllUserByCustomProperty(Ektron.Cms.Common.EkEnumeration.ObjectPropertyValueTypes PropType,
String PropValues);
```

Visual Basic

```
Public Function GetAllUserByCustomProperty(ByVal PropType As
Ektron.Cms.Common.EkEnumeration.ObjectPropertyValueTypes, ByVal PropValues As String) As
UserData()
```

Parameters

Parameters	Description
ByVal PropType As Ektron.Cms.Common.EkEnumeration.ObjectPropertyValueTypes	
ByVal PropValues As String	Required property

GetAllUserGroups Method

This is the overview for the GetAllUserGroups method overload.

Description

Loads all of the user groups from the system. This method is overloaded.

Overload List

	Name	Description
	User.GetAllUserGroups (Ektron.Cms.Common.EkEnumeration.UserTypes, String) (see page 970)	Loads all of the user groups from the system. This function is overloaded. See Also: GetAllUserGroups (String)
	User.GetAllUserGroups (String) (see page 971)	Loads all of the user groups from the system. This function is overloaded. See Also: GetAllUserGroups (EkEnumeration.UserTypes String)

User.GetAllUserGroups Method (Ektron.Cms.Common.EkEnumeration.UserTypes, String)

Loads all of the user groups from the system.

This function is overloaded. See Also: GetAllUserGroups (String)

C#

```
public UserGroupData[] GetAllUserGroups(Ektron.Cms.Common.EkEnumeration.UserTypes GroupType,
String OrderBy);
```

Visual Basic

```
Public Function GetAllUserGroups(ByVal GroupType As Ektron.Cms.Common.EkEnumeration.UserTypes,
ByVal OrderBy As String) As UserGroupData()
```

Parameters

Parameters	Description
ByVal GroupType As Ektron.Cms.Common.EkEnumeration.UserTypes	GroupType: 1 = Membership (see page 360); 0 = CMSUser
ByVal OrderBy As String	Arranges data in a specific order. Possible values are **GroupName** or **GroupID**.

Returns

Ektron.Cms.UserGroupData()

User.GetAllUserGroups Method (String)

Loads all of the user groups from the system.

This function is overloaded. See Also: GetAllUserGroups (EkEnumeration.UserTypes String)

C#

```
public UserGroupData[] GetAllUserGroups(String OrderBy);
```

Visual Basic

```
Public Function GetAllUserGroups(ByVal OrderBy As String) As UserGroupData()
```

Parameters

Parameters	Description
ByVal OrderBy As String	Arranges data in a specific order. Possible values are **GroupName** or **GroupID**.

Returns

Ektron.Cms.UserGroupData()

GetAllUsers Method

This is the overview for the GetAllUsers method overload.

Description

This method is overloaded.

Overload List

	Name	Description
	User.GetAllUsers (Long, Long) (see page 971)	Gets the User (see page 958) Group Data of all the allowed users in the folder specified.
	User.GetAllUsers (UserRequestData) (see page 972)	Returns the list of users from the database based on the UserRequestData class.

User.GetAllUsers Method (Long, Long)

Gets the User (see page 958) Group Data of all the allowed users in the folder specified.

C#

```
public UserGroupData[] GetAllUsers(Long CallerID, Long FolderId);
```

Visual Basic

```
Public Function GetAllUsers(ByVal CallerID As Long, ByVal FolderId As Long) As UserGroupData()
```

Parameters

Parameters	Description
ByVal CallerID As Long	The user ID that is logged on to the system.
ByVal FolderId As Long	The ID of the folder.

Returns

Ektron.Cms.UserGroupData()

User.GetAllUsers Method (UserRequestData)

Returns the list of users from the database based on the UserRequestData class.

C#

```
public UserData[] GetAllUsers(ref UserRequestData userRequest);
```

Visual Basic

```
Public Function GetAllUsers(ByRef userRequest As UserRequestData) As UserData()
```

Parameters

Parameters	Description
ByRef userRequest As UserRequestData	list of constraints to load users

Returns

An array of users as UserData.

User.GetGroupsUserIsIn Method

Loads all the groups by user.

C#

```
public GroupData[] GetGroupsUserIsIn(Long UserId, String OrderBy);
```

Visual Basic

```
Public Function GetGroupsUserIsIn(ByVal UserId As Long, ByVal OrderBy As String) As GroupData()
```

Parameters

Parameters	Description
ByVal UserId As Long	The user's ID.
ByVal OrderBy As String	The order in which the groups are displayed. The possible values are "GroupName" and "GroupID". (Required)

Returns

Ektron.Cms.GroupData()

User.GetMembershipUser Method

Returns user data based on the user's login name (username).

C#

```
public UserData GetMembershipUser(String Username);
```

Visual Basic

```
Public Function GetMembershipUser(ByVal Username As String) As UserData
```

Parameters

Parameters	Description
ByVal Username As String	User (see page 958)'s login name (username). This is required

Returns

Ektron.Cms.UserData

GetNonVerifyUser Method

This is the overview for the GetNonVerifyUser method overload.

Overload List

	Name	Description
	User.GetNonVerifyUser (String) (see page 973)	Returns an array of user group data based on the username
	User.GetNonVerifyUser (String, Boolean) (see page 973)	Returns an array of user group data based on the username , userpreference

User.GetNonVerifyUser Method (String)

Returns an array of user group data based on the username

C#

```
public UserData GetNonVerifyUser(String username);
```

Visual Basic

```
Public Function GetNonVerifyUser(ByVal username As String) As UserData
```

Parameters

Parameters	Description
ByVal username As String	The user's name. (Required)

Returns

Ektron.Cms.UserGroupData()

User.GetNonVerifyUser Method (String, Boolean)

Returns an array of user group data based on the username , userpreference

C#

```
public UserData GetNonVerifyUser(String username, Boolean Preference);
```

Visual Basic

```
Public Function GetNonVerifyUser(ByVal username As String, ByVal Preference As Boolean) As
UserData
```

Parameters

Parameters	Description
ByVal username As String	The user's name.(Required)
ByVal Preference As Boolean	Whether user preferences need to be returned (Required - Boolean)

Returns

Ektron.Cms.UserGroupData()

User.GetUser Method

Loads the user details for the given user.

C#

```
public UserData GetUser(Long Id, Optional ByVal Boolean Preference, Optional ByVal Boolean
```

```
ReturnDeleted);
```

Visual Basic

```
Public Function GetUser(ByVal Id As Long, Optional ByVal Preference As Boolean = False,
Optional ByVal ReturnDeleted As Boolean = False) As UserData
```

Parameters

Parameters	Description
ByVal Id As Long	The user ID.
Optional ByVal Preference As Boolean = False	Loads preference only if required (true = load preferences as a part of user data).

Returns

Ektron.Cms.UserData

User.GetUserByUsername Method

Loads the user details for the given user.

C#

```
public UserData GetUserByUsername(String username);
```

Visual Basic

```
Public Function GetUserByUsername(ByVal username As String) As UserData
```

Parameters

Parameters	Description
ByVal username As String	The username.

Returns

Ektron.Cms.UserData

Example

The following example shows how to create a Web page that retrieves user details for a given username. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetUserByUsername method. This method uses the InternalAdmin to retrieve the user details. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a label to show where to enter a username.
```
<asp:Label ID="Label1" runat="server" Text="Username: "></asp:Label>
```
2. Add a text box to enter the username.
```
<asp:TextBox ID="txtUserName" runat="server" Width="63px"></asp:TextBox>
```
3. add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the username box.
```
<asp:RequiredFieldValidator ID="rfvUserId" runat="server" ControlToValidate="txtUserName"
  ErrorMessage="Please enter Username"></asp:RequiredFieldValidator>
```
4. Add two breaks for spacing purposes.
```
<br /><br />
```
5. Add a button to submit the username.
```
<asp:Button ID="btnGetUser" runat="server" Text="Get Custom Properties" Width="148px" />
```
6. Add two more breaks for spacing purposes.
```
<br /><br />
```
7. Add a label to display the given user's details.
```
<asp:Label ID="lblResult" runat="server"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add button click event.

```
        Protected Sub btnGetUser_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGetUser.Click
```

2. Create an object for the user API (see page 669).

```
Dim userAPI As New Ektron.Cms.API.User.User
```

3. Create an object for the user data.

```
Dim userObj As Ektron.Cms.UserData
```

4. Create a Try/Catch around the GetUserByUsername method. This allows you to catch any exceptions and helps when debugging.

```
Try
    userObj = userAPI.GetUserByUsername(txtUserName.Text.Trim)
```

5. Check to make sure the a user ID exists.

```
    If userObj.Id = 0 Then
       lblResult.Text = "Invalid Username"
       Exit Sub
    End If
```

6. Create an integer place holder.

```
    Dim i As Integer
```

7. Check to make sure the attribute list is populated.

```
    If Not userObj.AttributeList Is Nothing Then
```

8. If the list is populated, display the results.

```
       lblResult.Text = "<table> <th>Name</th> <th>Value</th> <th>Type</th>"
       For i = 0 To userObj.AttributeList.Length - 1 Step 1
           lblResult.Text &= "<tr>"
           lblResult.Text &= "<td>" & userObj.AttributeList(i).Name & "</td>"
           lblResult.Text &= "<td>" & userObj.AttributeList(i).Value & "</td>"
           lblResult.Text &= "<td>" &
userObj.AttributeList(i).ValueType.ToString() & "</td>"
           lblResult.Text &= "</tr>"
       Next
       lblResult.Text &= "</table>"
```

9. If nothing is contained in the list, display a message.

```
    Else
       lblResult.Text &= "No custom properties set-up for this user"
   End If
```

10. If there is an error, let the user know what happened.

```
Catch ex As Exception
   lblResult.Text = ex.Message
End Try
End Sub
```

User.GetUserCustomProperties Method

Returns an array of custom properties for a given user's ID.

C#

```
public CustomAttribute[] GetUserCustomProperties(Long UserId);
```

Visual Basic

```
Public Function GetUserCustomProperties(ByVal UserId As Long) As CustomAttribute()
```

Parameters

Parameters	Description
ByVal UserId As Long	User (see page 958)'s ID

Returns

Ektron.Cms.CustomAttribute()

User.GetUserEmailInfo Method

Returns user data for a given user's ID.

C#

```
public UserData GetUserEmailInfo(Long Id);
```

Visual Basic

```
Public Function GetUserEmailInfo(ByVal Id As Long) As UserData
```

Parameters

Parameters	Description
ByVal Id As Long	The user's ID.

Returns

Object of UserData

User.GetUserForFolderAdmin Method

C#

```
public UserData GetUserForFolderAdmin(Long FolderId, Long Id, Optional ByVal Boolean
Preference, Optional ByVal Boolean ReturnDeleted);
```

Visual Basic

```
Public Function GetUserForFolderAdmin(ByVal FolderId As Long, ByVal Id As Long, Optional ByVal
Preference As Boolean = False, Optional ByVal ReturnDeleted As Boolean = False) As UserData
```

Parameters

Parameters	Description
ByVal FolderId As Long	
ByVal Id As Long	
Optional ByVal Preference As Boolean = False	

User.GetUserGroup Method

Loads a given group's details.

C#

```
public UserGroupData GetUserGroup(Long Id);
```

Visual Basic

```
Public Function GetUserGroup(ByVal Id As Long) As UserGroupData
```

Parameters

Parameters	Description
ByVal Id As Long	The group's ID.

Returns

Ektron.Cms.UserGroupData

User.GetUserGroupByName Method

Loads a given group's details.

C#

```
public UserGroupData GetUserGroupByName(String name);
```

Visual Basic

```
Public Function GetUserGroupByName(ByVal name As String) As UserGroupData
```

Parameters

Parameters	Description
ByVal name As String	The group's name.

Returns

Ektron.Cms.UserGroupData

User.GetUserGroupForFolderAdmin Method

Returns all the group's data for the administrator.

C#

```
public UserGroupData GetUserGroupForFolderAdmin(Long FolderId, Long Id);
```

Visual Basic

```
Public Function GetUserGroupForFolderAdmin(ByVal FolderId As Long, ByVal Id As Long) As
UserGroupData
```

Parameters

Parameters	Description
ByVal FolderId As Long	Folder (see page 683) ID
ByVal Id As Long	User (see page 958) ID

Returns

UserGroupData

User.GetUserPermissions Method

Loads the user permissions as a array of UserPermissionData.

C#

```
public UserPermissionData[] GetUserPermissions(Long Id, String ItemType, Long UserId, String
UserGroupList, Optional ByVal PermissionUserType PermissionType, Optional ByVal
PermissionRequestType PermissionRequest);
```

Visual Basic

```
Public Function GetUserPermissions(ByVal Id As Long, ByVal ItemType As String, ByVal UserId As
Long, ByVal UserGroupList As String, Optional ByVal PermissionType As PermissionUserType =
PermissionUserType.All, Optional ByVal PermissionRequest As PermissionRequestType =
PermissionRequestType.All) As UserPermissionData()
```

Parameters

Parameters	Description
ByVal Id As Long	The item's ID (content or folder id whose permissions you are seeking).
ByVal ItemType As String	The item's type (valid values - content or folder).
ByVal UserId As Long	The user's ID (when you are getting permissions for a group set this to zero).
ByVal UserGroupList As String	The group list (comma separated list of group ids, blank when you are getting permissions for a specific user).
Optional ByVal PermissionType As PermissionUserType = PermissionUserType.All	PermissionUserType (Use a value from enum Ektron.Cms.Content.EkContent.PermissionResultType).

Optional ByVal PermissionRequest As PermissionRequestType = PermissionRequestType.All	PermissionRequestType (Use a value from enum Ektron.Cms.Content.EkContent.PermissionRequestType).

Returns

Ektron.Cms.UserPermissionData()

User.GetUserPreference Method

Loads the preferences for the given user.

C#

```
public UserPreferenceData GetUserPreference(Long Id);
```

Visual Basic

```
Public Function GetUserPreference(ByVal Id As Long) As UserPreferenceData
```

Parameters

Parameters	Description
ByVal Id As Long	The user's ID.

Returns

Ektron.Cms.UserPreferenceData

GetUsers Method

This is the overview for the GetUsers method overload.

Description

An Overloaded function that allows you to get the users from a specific group or load and array of user group data. This method is overloaded.

Overload List

	Name	Description
	User.GetUsers (Long, String) (see page 978)	Loads the all the users from the given group.
	User.GetUsers (String, String) (see page 979)	Returns an array of user group data based on the userName and userNameType.

User.GetUsers Method (Long, String)

Loads the all the users from the given group.

C#

```
public UserData[] GetUsers(Long GroupId, String OrderBy);
```

Visual Basic

```
Public Function GetUsers(ByVal GroupId As Long, ByVal OrderBy As String) As UserData()
```

Parameters

Parameters	Description
ByVal GroupId As Long	The group's ID

ByVal OrderBy As String	The order of the users. The string value can be: • username • firstname • lastname • userid • language • datecreated • lastlogindate

Returns

- Ektron.Cms.UserData()

Description

User.GetUsers Method (String, String)

Returns an array of user group data based on the userName and userNameType.

C#

```
public UserGroupData[] GetUsers(String userNames, Optional ByVal String userNameTypes);
```

Visual Basic

```
Public Function GetUsers(ByVal userNames As String, Optional ByVal userNameTypes As String =
"") As UserGroupData()
```

Parameters

Parameters	Description
ByVal userNames As String	The user's name. (Required)
Optional ByVal userNameTypes As String = ""	The user type possible values are **user** or **membership user**.

Returns

Ektron.Cms.UserGroupData()

User.GetUsersForTask Method

Loads the users that have permissions to the given content.

C#

```
public UserGroupData[] GetUsersForTask(Long CallerID, Long ContentID);
```

Visual Basic

```
Public Function GetUsersForTask(ByVal CallerID As Long, ByVal ContentID As Long) As
UserGroupData()
```

Parameters

Parameters	Description
ByVal CallerID As Long	The user ID that is logged on to the system.
ByVal ContentID As Long	The ID of the content. (Required)

Returns

Ektron.Cms.UserGroupData()

User.GetUsersNotInGroup Method

Loads all the users that are not in the given group.

C#

```
public UserData[] GetUsersNotInGroup(Long Id, String OrderBy, Optional ByVal int GroupType);
```

Visual Basic

```
Public Function GetUsersNotInGroup(ByVal Id As Long, ByVal OrderBy As String, Optional ByVal
GroupType As Integer = 0) As UserData()
```

Parameters

Parameters	Description
ByVal Id As Long	The group ID.
ByVal OrderBy As String	The order of the users. The string value can be: • username • firstname • lastname • userid • language • datecreated • lastlogindate

Returns

Ektron.Cms.UserData()

Description

User.IsAdmin Method

Determines the if the current user is an Administrator.

C#

```
public Boolean IsAdmin();
```

Visual Basic

```
Public Function IsAdmin() As Boolean
```

Returns

True if the logged in user is administrator. Otherwise, returns false.

User.IsAGroupMember Method

For a given userId, returns whether that user is a member of the group.

C#

```
public Boolean IsAGroupMember(Long userId, Long groupId);
```

Visual Basic

```
Public Function IsAGroupMember(ByVal userId As Long, ByVal groupId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal userId As Long	Long value of the user Id.
ByVal groupId As Long	Long value of the group Id.

Returns

Boolean of whether the user is a member of the group.

IsARoleMember Method

This is the overview for the IsARoleMember method overload.

Description

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This method is overloaded.

Overload List

	Name	Description
	User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) (see page 982)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This function is overloaded. See Also: • IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Long Boolean) • IsARoleMember (Long Long Boolean) • IsARoleMember (Long)
	User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Long, Boolean) (see page 982)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This function is overloaded. See Also: • IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) • IsARoleMember (Long Long Boolean) • IsARoleMember (Long)
	User.IsARoleMember (Long) (see page 983)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This function is overloaded. See Also: • IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) • IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Long Boolean) • IsARoleMember (Long Long Boolean)

	User.IsARoleMember (Long, Long, Boolean) (see page 983)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This function is overloaded. See Also: • IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Long Boolean) • IsARoleMember (Long) • IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)

User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Long Boolean)
- IsARoleMember (Long Long Boolean)
- IsARoleMember (Long)

C#

```
public Boolean IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId);
```

Visual Basic

```
Public Function IsARoleMember(ByVal RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds) As
Boolean
```

Parameters

Parameters	Description
ByVal RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds	_nt_

Remarks

User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Long, Boolean)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)
- IsARoleMember (Long Long Boolean)
- IsARoleMember (Long)

C#

```
public Boolean IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId, Long nUserId,
Optional ByVal Boolean GroupFlag);
```

Visual Basic

```
Public Function IsARoleMember(ByVal RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds,
```

```
ByVal nUserId As Long, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
ByVal RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds	_nt_
ByVal nUserId As Long	_nt_
Optional ByVal GroupFlag As Boolean = False	_nt_

Remarks

User.IsARoleMember Method (Long)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)
- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Long Boolean)
- IsARoleMember (Long Long Boolean)

C#

```
public Boolean IsARoleMember(Long RoleId);
```

Visual Basic

```
Public Function IsARoleMember(ByVal RoleId As Long) As Boolean
```

Parameters

Parameters	Description
ByVal RoleId As Long	_nt_

Remarks

User.IsARoleMember Method (Long, Long, Boolean)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Long Boolean)
- IsARoleMember (Long)
- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)

C#

```
public Boolean IsARoleMember(Long RoleId, Long nUserId, Optional ByVal Boolean GroupFlag);
```

Visual Basic

```
Public Function IsARoleMember(ByVal RoleId As Long, ByVal nUserId As Long, Optional ByVal
GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
ByVal RoleId As Long	_nt_

ByVal nUserId As Long	_nt_
Optional ByVal GroupFlag As Boolean = False	_nt_

Remarks

User.IsARoleMemberForContent Method

Determines whether a Role Member is allowed to work with a given piece of content as an administrator or not. Administrations privileges are for the given piece of content only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

C#

```
public Boolean IsARoleMemberForContent(Long RoleId, Long ContentId, Optional ByVal Long
nUserId, Optional ByVal Boolean GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForContent(ByVal RoleId As Long, ByVal ContentId As Long,
Optional ByVal nUserId As Long = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
ByVal RoleId As Long	
ByVal ContentId As Long	The numeric content ID for which you are giving the Role Member privileges.
Optional ByVal nUserId As Long = 0	
Optional ByVal GroupFlag As Boolean = False	

User.IsARoleMemberForContent_FolderUserAdmin Method

Determines whether a Role Member is allowed to work with a given conatent as an administrator or not. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

C#

```
public Boolean IsARoleMemberForContent_FolderUserAdmin(Long ContentId, Optional ByVal Long
nUserId, Optional ByVal Boolean GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForContent_FolderUserAdmin(ByVal ContentId As Long, Optional
ByVal nUserId As Long = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
ByVal ContentId As Long	required content id
Optional ByVal nUserId As Long = 0	user id
Optional ByVal GroupFlag As Boolean = False	flag

Returns

True if granted otherwise false

User.IsARoleMemberForFolder Method

Determines whether a Role Member is allowed to work with a given folder as an administrator or not. Administrations privileges are for the given folder only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

C#

```
public Boolean IsARoleMemberForFolder(Long RoleId, Long FolderId, Optional ByVal Long nUserId,
Optional ByVal Boolean GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForFolder(ByVal RoleId As Long, ByVal FolderId As Long, Optional
ByVal nUserId As Long = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
ByVal RoleId As Long	
ByVal FolderId As Long	The numeric folder ID for which you are giving the Role Member privileges.
Optional ByVal nUserId As Long = 0	
Optional ByVal GroupFlag As Boolean = False	

Returns

True if granted otherwise false

User.IsARoleMemberForFolder_FolderUserAdmin Method

Determines whether a Role Member is allowed to work with a given folder as an administrator or not. Administrations privileges are for the given folder only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

C#

```
public Boolean IsARoleMemberForFolder_FolderUserAdmin(Long FolderId, Optional ByVal Long
nUserId, Optional ByVal Boolean GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForFolder_FolderUserAdmin(ByVal FolderId As Long, Optional ByVal
nUserId As Long = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
ByVal FolderId As Long	required folder id
Optional ByVal nUserId As Long = 0	user id
Optional ByVal GroupFlag As Boolean = False	group id

Returns

True if granted otherwise false

User.IsARoleMemberForFolder_MetadataAdmin Method

Determines whether a Role Member is allowed to work with a given folder as an meta data administrator or not.

C#

```
public Boolean IsARoleMemberForFolder_MetadataAdmin(Long FolderId, Optional ByVal Long
nUserId, Optional ByVal Boolean GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForFolder_MetadataAdmin(ByVal FolderId As Long, Optional ByVal
nUserId As Long = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
ByVal FolderId As Long	required folder id

Optional ByVal nUserId As Long = 0	user id
Optional ByVal GroupFlag As Boolean = False	group id

Returns

True if granted otherwise false

User.LockUserAccount Method

This function locks a user account.

C#

```
public LockUserAccount(Long UserId);
```

Visual Basic

```
Public Sub LockUserAccount(ByVal UserId As Long)
```

Parameters

Parameters	Description
ByVal UserId As Long	A unique ID for the required user.

LogInUser Method

This is the overview for the LogInUser method overload.

Overload List

	Name	Description
	User.LogInUser (String, String, String) (see page 986)	This method allows you to log a user into a server by passing their username and password.
	User.LogInUser (String, String, String, String) (see page 987)	This method allows you to log a user into a server by passing their username, password, and server name.
	User.LogInUser (String, String, String, String, EkEnumeration.AutoAddUserTypes) (see page 987)	Used to force log in the user and no password required.
	User.LogInUser (String, String, String, String, String) (see page 988)	This method allows you to log a user into a server by passing their username, password, and server name. You can also log in an Active Directory (see page 196) user by providing the username, password and domain. Note: When when using Active Directory (see page 196), the ServerName and Protocol parameters are set to empty strings ("").

User.LogInUser Method (String, String, String)

This method allows you to log a user into a server by passing their username and password.

C#

```
public UserData LogInUser(String username, String Password, String ServerName);
```

Visual Basic

```
Public Function LogInUser(ByVal username As String, ByVal Password As String, ByVal ServerName
As String) As UserData
```

Parameters

Parameters	Description
ByVal username As String	The system username.
ByVal Password As String	The system password.
ByVal ServerName As String	The system server name. When using Active Directory (see page 196), this parameter needs to be an empty string "".

Returns

Ektron.Cms.UserData

User.LogInUser Method (String, String, String, String)

This method allows you to log a user into a server by passing their username, password, and server name.

C#

```
public UserData LogInUser(String username, String Password, String ServerName, String Domain);
```

Visual Basic

```
Public Function LogInUser(ByVal username As String, ByVal Password As String, ByVal ServerName
As String, ByVal Domain As String) As UserData
```

Parameters

Parameters	Description
ByVal username As String	The system username.
ByVal Password As String	The system password.
ByVal ServerName As String	The system server name. When using Active Directory (see page 196), this parameter needs to be an empty string "".
ByVal Domain As String	The user's domain, required only when users login using Active Directory (see page 196). If your domain is **intra.example.com**, it is entered in the following format: **"DC=intra,DC=example,DC=com"**.

Returns

Ektron.Cms.UserData

User.LogInUser Method (String, String, String, String, EkEnumeration.AutoAddUserTypes)

Used to force log in the user and no password required.

C#

```
public UserData LogInUser(String username, String ServerName, String Domain, String Protocol,
EkEnumeration.AutoAddUserTypes AutoAddType);
```

Visual Basic

```
Public Function LogInUser(ByVal username As String, ByVal ServerName As String, ByVal Domain
As String, ByVal Protocol As String, ByVal AutoAddType As EkEnumeration.AutoAddUserTypes) As
UserData
```

Parameters

Parameters	Description
ByVal username As String	The system username.
ByVal ServerName As String	The system server name. When using Active Directory (see page 196), this parameter needs to be an empty string "".
ByVal Domain As String	The user's domain, required only when users login using Active Directory (see page 196). If your domain is **intra.example.com**, it is entered in the following format: **"DC=intra,DC=example,DC=com"**.
ByVal Protocol As String	Required only when using a secure login or when you are using any valid Active Directory (see page 196) protocols.
ByVal AutoAddType As EkEnumeration.AutoAddUserTypes	The desired EkEnumeration.AutoAddUserTypes type

Returns

Ektron.Cms.UserData

Remarks

Using this method may increase security risk

Notes

When using Active Directory (see page 196)/LDAP, this method will log the user only after first successfull LDAP/AD User (see page 958) Login (see page 347) into CMS. when using Active Directory (see page 196), the ServerName and Protocol parameters are set to empty strings ("").

User.LogInUser Method (String, String, String, String, String)

This method allows you to log a user into a server by passing their username, password, and server name. You can also log in an Active Directory (see page 196) user by providing the username, password and domain. Note: When when using Active Directory (see page 196), the ServerName and Protocol parameters are set to empty strings ("").

C#

```
public UserData LogInUser(String username, String Password, String ServerName, String Domain,
String Protocol);
```

Visual Basic

```
Public Function LogInUser(ByVal username As String, ByVal Password As String, ByVal ServerName
As String, ByVal Domain As String, ByVal Protocol As String) As UserData
```

Parameters

Parameters	Description
ByVal username As String	The system username.
ByVal Password As String	The system password.
ByVal ServerName As String	The system server name. When using Active Directory (see page 196), this parameter needs to be an empty string "".
ByVal Domain As String	The user's domain, required only when users login using Active Directory (see page 196). If your domain is **intra.example.com**, it is entered in the following format: **"DC=intra,DC=example,DC=com"**.
ByVal Protocol As String	This property is always set to an empty string "". This parameter is only required when using Active Directory (see page 196).

Returns

Ektron.Cms.UserData

Example

The following example shows how to create a Web page that programmatically logs a user into CMS400.NET. Please note that the code-behind in this example does not login a new user if someone is already logged into the Workarea. This example uses some standard drag and drop controls and a small section of VB code utilizing the LogInUser method. This method uses the InternalAdmin to log the user into CMS400.NET. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a label to show where the Username is entered.
```
<asp:Label ID="Label6" runat="server" Text="Username: " Width="76px"></asp:Label>
```
2. Add a text box to enter the Username.
```
<asp:TextBox ID="txtUserName" runat="server" Width="134px"></asp:TextBox>
```
3. Add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the Username box.
```
<asp:RequiredFieldValidator ID="rfvUserName" runat="server" ErrorMessage="Please enter user
name"
    ControlToValidate="txtUserName">Please enter user name</asp:RequiredFieldValidator>
```
4. Add two breaks for spacing purposes.

```
<br /><br />
```

5. Add a label to show where the Password is entered.

```
<asp:Label ID="Label8" runat="server" Text="Password: " Width="76px"></asp:Label>
```

6. Add a text box to enter the Password.

```
<asp:TextBox ID="txtPassword" runat="server" TextMode="Password"
Width="134px"></asp:TextBox>
```

7. Add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the password box.

```
<asp:RequiredFieldValidator ID="rfvPassword" runat="server" ErrorMessage="Please enter
password"
  ControlToValidate="txtPassword">Please enter password</asp:RequiredFieldValidator>
```

8. Add two breaks for spacing purposes.

```
<br /><br />
```

9. Add a button to submit the user's login information. We will add a button click event in the code behind.

```
<asp:Button ID="btnLogin" runat="server" Text="Login" Width="57px" />
```

10. Add two last breaks for spacing purposes.

```
<br /><br />
```

11. Add a results label to display a message that states whether the user is logged in.

```
<asp:Label ID="lblResult" runat="server" Width="432px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a page load event that checks to see if a user is logged into CMS400.NET.

```
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
        Dim api As New Ektron.Cms.CommonApi
        If (api.RequestInformationRef.CallerId > 0) Then
            lblResult.Text = api.RequestInformationRef.LoggedInUsername & " is logged in."
& <"br/>"
        End If
    End Sub
```

2. Add a button click event.

```
    Protected Sub btnLogin_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnLogin.Click
```

3. Create an object for the user API (see page 669).

```
        Dim userAPI As New Ektron.Cms.API.User.User
```

4. Create an object for the common API (see page 669).

```
        Dim api As New Ektron.Cms.CommonApi
```

5. Create an object to store the user data.

```
        Dim userObj As UserData
```

6. Create a Try/Catch around the LogInUser method. This allows you to catch any exceptions and helps when debugging.

7. Check to see if the user is already logged in.

```
        Try
            If (api.RequestInformationRef.CallerId > 0) Then
                lblResult.Text = api.RequestInformationRef.LoggedInUsername & " is already
logged in."
```

8. If the user is not logged in, proceed with the login.

```
            Else
                userObj = userAPI.LogInUser(txtUserName.Text.Trim, txtPassword.Text.Trim,
"localhost", "", "")
```

9. Check to see whether the login was successful. If it was unsuccessful, display a message.

```
                If userObj.Id = 0 Then
                    lblResult.Text = "Login failed."
                    Exit Sub
                End If
```

10. Set the authentication cookie and Response.Redirect.

```
                    userAPI.SetAuthenticationCookie(userObj)
                    Response.Redirect(Request.Url.ToString)
                End If
```

11. If there is an error, let the user know what happened.

```
            Catch ex As Exception
                lblResult.Text = ex.Message
            End Try
        End Sub
```

User.ReadCustomProperties Method

Returns the custom properties as a Visual Basic collection. Importing the Microsoft.VisualBasic name space might be required.

C#

```
public Collection ReadCustomProperties(Collections.Specialized.NameValueCollection form);
```

Visual Basic

```
Public Function ReadCustomProperties(ByVal form As
Collections.Specialized.NameValueCollection) As Collection
```

Parameters

Parameters	Description
ByVal form As Collections.Specialized.NameValueCollection	A collection of user properties.

Returns

Microsoft.VisualBasic.Collection (see page 95)

ResetMembershipUserPassword Method

This is the overview for the ResetMembershipUserPassword method overload.

Overload List

	Name	Description
	User.ResetMembershipUserPassword (String) (see page 990)	Reset the password for the given membership user.
	User.ResetMembershipUserPassword (String, String) (see page 991)	Reset the password for the given membership user.

User.ResetMembershipUserPassword Method (String)

Reset the password for the given membership user.

C#

```
public String ResetMembershipUserPassword(String Username);
```

Visual Basic

```
Public Function ResetMembershipUserPassword(ByVal Username As String) As String
```

Parameters

Parameters	Description
ByVal Username As String	User (see page 958)'s username to be reset.

Returns

String

User.ResetMembershipUserPassword Method (String, String)

Reset the password for the given membership user.

C#

```
public String ResetMembershipUserPassword(String Username, String AccountId);
```

Visual Basic

```
Public Function ResetMembershipUserPassword(ByVal Username As String, ByVal AccountId As
String) As String
```

Parameters

Parameters	Description
ByVal Username As String	User (see page 958)'s username to be reset.
ByVal AccountId As String	If verification is turned on, specify verification id(email to a user).

Returns

String

User.SetAuthenticationCookie Method

This method sets the authentication cookie.

C#

```
public SetAuthenticationCookie(UserData userInfo);
```

Visual Basic

```
Public Sub SetAuthenticationCookie(ByVal userInfo As UserData)
```

Parameters

Parameters	Description
ByVal userInfo As UserData	Returned value of the logInUser method.

Remarks

First, you must call the logInUser method to login the user.

User.UnsubscribeUser Method

Unsubscribes a user from the system.

C#

```
public Boolean UnsubscribeUser(String Username, Optional ByVal String Password);
```

Visual Basic

```
Public Function UnsubscribeUser(ByVal Username As String, Optional ByVal Password As String =
"") As Boolean
```

Parameters

Parameters	Description
ByVal Username As String	The user's username.
Optional ByVal Password As String = ""	The user's password.

Returns

Boolean

User.UpdateCustomPropertiesItemOrder Method

Updates the order of available custom properties.

C#

```
public Boolean UpdateCustomPropertiesItemOrder(String ItemOrder);
```

Visual Basic

```
Public Function UpdateCustomPropertiesItemOrder(ByVal ItemOrder As String) As Boolean
```

Parameters

Parameters	Description
ByVal ItemOrder As String	Order of the properties applicable to the user.

Returns

Boolean

UpdateUser Method

This is the overview for the UpdateUser method overload.

Overload List

	Name	Description
	User.UpdateUser (Collection) (see page 992)	Updates the user details.
	User.UpdateUser (UserData) (see page 992)	Updates the user details. If you do not wish to update the user's password, leave the UserData.Password property blank.

User.UpdateUser Method (Collection)

Updates the user details.

C#

```
public UpdateUser(Collection data);
```

Visual Basic

```
Public Sub UpdateUser(ByVal data As Collection)
```

Parameters

Parameters	Description
ByVal data As Collection	A collection of user data.

User.UpdateUser Method (UserData)

Updates the user details. If you do not wish to update the user's password, leave the UserData.Password property blank.

C#

```
public UpdateUser(UserData data);
```

Visual Basic

```
Public Sub UpdateUser(ByVal data As UserData)
```

Parameters

Parameters	Description
ByVal data As UserData	A UserData object of user data.

UpDateUserGroup Method

This is the overview for the UpDateUserGroup method overload.

Overload List

	Name	Description
	User.UpDateUserGroup (Collection) (see page 993)	Updates the user group details.
	User.UpDateUserGroup (UserData) (see page 993)	Updates the user group details.

User.UpDateUserGroup Method (Collection)

Updates the user group details.

C#

```
public UpDateUserGroup(Collection Group);
```

Visual Basic

```
Public Sub UpDateUserGroup(ByVal Group As Collection)
```

Parameters

Parameters	Description
ByVal Group As Collection	A list of group details as a collection.

User.UpDateUserGroup Method (UserData)

Updates the user group details.

C#

```
public UpDateUserGroup(UserData data);
```

Visual Basic

```
Public Sub UpDateUserGroup(ByVal data As UserData)
```

Parameters

Parameters	Description
ByVal data As UserData	data As UserData)

User.UpdateUserPreferences Method

Updates the given user's preference data.

C#

```
public UpdateUserPreferences(Long Id, Collection data);
```

Visual Basic

```
Public Sub UpdateUserPreferences(ByVal Id As Long, ByVal data As Collection)
```

Parameters

Parameters	Description
ByVal Id As Long	The user's ID.
ByVal data As Collection	A collection of user preferences.

UserObject Method

This is the overview for the UserObject method overload.

Description

Returns user data based on the user's ID or by the user's login or username. This function is overloaded.

Overload List

	Name	Description
	User.UserObject (Long) (see page 994)	Returns user data based on the user's ID. This function is overloaded. See Also: UserObject (String)
	User.UserObject (String) (see page 994)	Returns user data based on the user's login or username. This function is overloaded. See Also: UserObject (Long)

User.UserObject Method (Long)

Returns user data based on the user's ID.

This function is overloaded. See Also: UserObject (String)

C#

```
public UserData UserObject(Long Id);
```

Visual Basic

```
Public Function UserObject(ByVal Id As Long) As UserData
```

Parameters

Parameters	Description
ByVal Id As Long	The user's ID.

Returns

Ektron.Cms.UserData

User.UserObject Method (String)

Returns user data based on the user's login or username.

This function is overloaded. See Also: UserObject (Long)

C#

```
public UserData UserObject(String sUserName);
```

Visual Basic

```
Public Function UserObject(ByVal sUserName As String) As UserData
```

Parameters

Parameters	Description
ByVal sUserName As String	User (see page 958)'s login or username.

Returns

Ektron.Cms.UserData

User Properties

The properties of the User class are listed here.

Public Properties

	Name	Description
	UserIdAuthenticated (see page 995)	Returns the id of the currently logged in user.

User.UserIdAuthenticated Property

Returns the id of the currently logged in user.

C#

```
public ReadOnly int UserIdAuthenticated;
```

Visual Basic

```
Public ReadOnly Property UserIdAuthenticated() As Integer
```

Returns

Boolean

Ektron.Cms.Framework Namespace

This is namespace Ektron.Cms.Framework.

Namespaces

Name	Description
Activity (see page 996)	This is namespace Ektron.Cms.Framework.Activity.
Analytics (see page 1005)	This is namespace Ektron.Cms.Framework.Analytics.
Calendar (see page 1037)	This is namespace Ektron.Cms.Framework.Calendar.
Core (see page 1047)	This is namespace Ektron.Cms.Framework.Core.
Messaging (see page 1052)	This is namespace Ektron.Cms.Framework.Messaging.
Notifications (see page 1060)	This is namespace Ektron.Cms.Framework.Notifications.
SocialNetworking (see page 1069)	This is namespace Ektron.Cms.Framework.SocialNetworking.
Users (see page 1076)	This is namespace Ektron.Cms.Framework.Users.

Classes

	Name	Description
	CmsApi (see page 1080)	Base API (see page 669) class for all Ektron CMS APIs.

Structs, Records, Enums

	Name	Description
	ApiAccessMode (see page 1082)	Enumeration used to indicate the access mode when using an API (see page 669) class.

Ektron.Cms.Framework.Activity Namespace

This is namespace Ektron.Cms.Framework.Activity.

Classes

	Name	Description
	Activity (see page 996)	This is class Ektron.Cms.Framework.Activity.Activity.
	ActivityQueue (see page 1001)	Api class for Queuing activities.
	ActivityStream (see page 1001)	Api class for managing activities in ActivityStreams.
	ActivityType (see page 1004)	Api class for managing Ektron.Cms (see page 7).

Classes

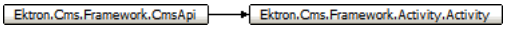
The following table lists classes in this documentation.

Classes

	Name	Description
	Activity (see page 996)	This is class Ektron.Cms.Framework.Activity.Activity.
	ActivityQueue (see page 1001)	Api class for Queuing activities.
	ActivityStream (see page 1001)	Api class for managing activities in ActivityStreams.
	ActivityType (see page 1004)	Api class for managing Ektron.Cms (see page 7).

Activity Class

Class Hierarchy



C#

```
public class Activity : CmsApi;
```

Visual Basic

```
Public Class Activity
Inherits CmsApi
```

File

Activity.cs

Description

This is class Ektron.Cms.Framework.Activity.Activity.

Activity.Activity Constructor

C#

```
public Activity(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

Activity Methods

The methods of the Activity class are listed here.

Public Methods

	Name	Description
	Add (see page 997)	Adds a new activity to the general feed.
	AddActivityMessage (see page 997)	Adds a language specific message for an activity in the general feed.
	Delete (see page 998)	Delete's an activity from the general feed.
	DisableActivityPublishing (see page 998)	Disables activity publishing for the site.
	EnableActivityPublishing (see page 998)	Enables activity publishing for the site. This will also stup defualt notification preferences for any existin users. Warning, depending on the number of users, this call could be time consuming.
	GetItem (see page 998)	Returns a single ActivityStream (see page 1001) Item by ID.
	GetList (see page 998)	This is the overview for the GetList method overload.
	GetListForUser (see page 999)	Returns a list of activities associated with a user. These will be activities where the user is the subject or the Actor.
	Publish (see page 1000)	Publishes an activity to all subscribers.
	Update (see page 1000)	Updates an activity in the general feed.
	UpdateActivityMessage (see page 1000)	Updates an existing message for an activity in the general feed.

Activity.Add Method

Adds a new activity to the general feed.

C#

```
public void Add(ActivityData activity);
```

Visual Basic

```
Public Function Add(activity As ActivityData) As void
```

Parameters

Parameters	Description
ActivityData activity	Activity (see page 996) data to add.

Activity.AddActivityMessage Method

Adds a language specific message for an activity in the general feed.

C#

```
public void AddActivityMessage(long activityId, int languageId, string message);
```

Visual Basic

```
Public Function AddActivityMessage(activityId As long, languageId As Integer, message As
string) As void
```

Parameters

Parameters	Description
long activityId	Id of activity to add message for.
int languageId	Id of language to add message for.
string message	Activity (see page 996) message to be added.

Activity.Delete Method

Delete's an activity from the general feed.

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	The ID of activity to delete.

Activity.DisableActivityPublishing Method

Disables activity publishing for the site.

C#

```
public void DisableActivityPublishing();
```

Visual Basic

```
Public Function DisableActivityPublishing() As void
```

Activity.EnableActivityPublishing Method

Enables activity publishing for the site. This will also stup defualt notification preferences for any existin users. Warning, depending on the number of users, this call could be time consuming.

C#

```
public void EnableActivityPublishing();
```

Visual Basic

```
Public Function EnableActivityPublishing() As void
```

Activity.GetItem Method

Returns a single ActivityStream (see page 1001) Item by ID.

C#

```
public ActivityData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As ActivityData
```

Parameters

Parameters	Description
long id	ID of Activtiy to retrieve.

GetList Method

This is the overview for the GetList method overload.

Overload List

	Name	Description
	Activity.GetList (Criteria<ActivityProperty>) (see page 999)	Returns a list of all activities based upon criteria supplied.
	Activity.GetList (Criteria<ActivityProperty>, EkEnumeration.ActivityFeedType, long) (see page 999)	Gets an Activity (see page 996) Feed based upon the feedtype and the Id.

Activity.GetList Method (Criteria<ActivityProperty>)

Returns a list of all activities based upon criteria supplied.

C#

```
public List<ActivityData> GetList(Criteria<ActivityProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<ActivityProperty>) As List<ActivityData>
```

Parameters

Parameters	Description
Criteria<ActivityProperty> criteria	Criteria used to filter results.

Activity.GetList Method (Criteria<ActivityProperty>, EkEnumeration.ActivityFeedType, long)

Gets an Activity (see page 996) Feed based upon the feedtype and the Id.

C#

```
public List<ActivityData> GetList(Criteria<ActivityProperty> criteria,
EkEnumeration.ActivityFeedType feedType, long feedTypeId);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<ActivityProperty>, feedType As
EkEnumeration.ActivityFeedType, feedTypeId As long) As List<ActivityData>
```

Parameters

Parameters	Description
Criteria<ActivityProperty> criteria	criteria for further filtering the feed, sorting, and paging.
EkEnumeration.ActivityFeedType feedType	The type of feed to retrieve. For example, User (see page 950) or CommunityGroup (see page 750).
long feedTypeId	The id corresponding to the feed type. For example, if FeedType = User (see page 950), FeedTypeId would be the user Id to retrieve feed for.

Activity.GetListForUser Method

Returns a list of activities associated with a user. These will be activities where the user is the subject or the Actor.

C#

```
public List<ActivityData> GetListForUser(long userId, PagingInfo pagingInfo);
```

Visual Basic

```
Public Function GetListForUser(userId As long, pagingInfo As PagingInfo) As List<ActivityData>
```

Parameters

Parameters	Description
long userId	ID of user to retrieve feed for.
PagingInfo pagingInfo	Paging information for retrieval.

Activity.Publish Method

Publishes an activity to all subscribers.

C#

```
public void Publish(ActivityData activity);
```

Visual Basic

```
Public Function Publish(activity As ActivityData) As void
```

Parameters

Parameters	Description
ActivityData activity	activity that has occured.

Activity.Update Method

Updates an activity in the general feed.

C#

```
public void Update(ActivityData activity);
```

Visual Basic

```
Public Function Update(activity As ActivityData) As void
```

Parameters

Parameters	Description
ActivityData activity	Activity (see page 996) data to update.

Activity.UpdateActivityMessage Method

Updates an existing message for an activity in the general feed.

C#

```
public void UpdateActivityMessage(long activityId, int languageId, string message);
```

Visual Basic

```
Public Function UpdateActivityMessage(activityId As long, languageId As Integer, message As
string) As void
```

Parameters

Parameters	Description
long activityId	Id of activity to add message for.
int languageId	Id of language to add message for.
string message	Activity (see page 996) message to be added.

Activity Properties

The properties of the Activity class are listed here.

Public Properties

	Name	Description
	IsActivityPublishingEnabled (see page 1001)	Gets setting indicating if Activity (see page 996) Publishing is enabled. Read Only - use EnableActivityPublishing (see page 998) and DisableActivityPublishing (see page 998) to set property.

Activity.IsActivityPublishingEnabled Property

Gets setting indicating if Activity (see page 996) Publishing is enabled. Read Only - use EnableActivityPublishing (see page 998) and DisableActivityPublishing (see page 998) to set property.

C#

```
public bool IsActivityPublishingEnabled;
```

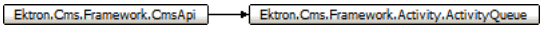
Visual Basic

```
Public Property IsActivityPublishingEnabled() As bool
```

ActivityQueue Class

Api class for Queuing activities.

Class Hierarchy



C#

```
public class ActivityQueue : CmsApi;
```

Visual Basic

```
Public Class ActivityQueue
Inherits CmsApi
```

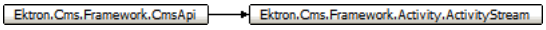
File

ActivityQueue.cs

ActivityStream Class

Api class for managing activities in ActivityStreams.

Class Hierarchy



C#

```
public class ActivityStream : CmsApi;
```

Visual Basic

```
Public Class ActivityStream
Inherits CmsApi
```

File

ActivityStream.cs

ActivityStream.ActivityStream Constructor

C#

```
public ActivityStream(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

ActivityStream Methods

The methods of the ActivityStream class are listed here.

Public Methods

	Name	Description
	GetUserActivityStream (see page 1002)	This is the overview for the GetUserActivityStream method overload.
	SendActivityToAllStreams (see page 1003)	Sends the activity directly to all user's streams.
	SendActivityToStreams (see page 1003)	Sends the activity to all applicable streams based upon the activity and users preferences.
	SendActivityToUsersStream (see page 1003)	Adds an activity existing in the general feed to a user's feed.

GetUserActivityStream Method

This is the overview for the GetUserActivityStream method overload.

Overload List

	Name	Description
	ActivityStream.GetUserActivityStream (Criteria<ActivityProperty>, long) (see page 1002)	Returns a list of activities in a user's ActivityStream (see page 1001) based upon supplied criteria.
	ActivityStream.GetUserActivityStream (long, PagingInfo) (see page 1002)	Returns a list of activities in a user's ActivityStream (see page 1001).

ActivityStream.GetUserActivityStream Method (Criteria<ActivityProperty>, long)

Returns a list of activities in a user's ActivityStream (see page 1001) based upon supplied criteria.

C#

```
public List<ActivityData> GetUserActivityStream(Criteria<ActivityProperty> criteria, long
userId);
```

Visual Basic

```
Public Function GetUserActivityStream(criteria As Criteria<ActivityProperty>, userId As long)
As List<ActivityData>
```

Parameters

Parameters	Description
Criteria<ActivityProperty> criteria	Criteira by which to filter the stream. UserId (see page 1082) filter is ignored.
long userId	ID of user to retrieve feed for.

ActivityStream.GetUserActivityStream Method (long, PagingInfo)

Returns a list of activities in a user's ActivityStream (see page 1001).

C#

```
public List<ActivityData> GetUserActivityStream(long userId, PagingInfo paging);
```

Visual Basic

```
Public Function GetUserActivityStream(userId As long, paging As PagingInfo) As
List<ActivityData>
```

Parameters

Parameters	Description
long userId	ID of user to retrieve feed for.
PagingInfo paging	Pagign Information for retrieval.

ActivityStream.SendActivityToAllStreams Method

Sends the activity directly to all user's streams.

C#

```
public void SendActivityToAllStreams(long activityId);
```

Visual Basic

```
Public Function SendActivityToAllStreams(activityId As long) As void
```

Parameters

Parameters	Description
long activityId	Id of activity to send to streams.

ActivityStream.SendActivityToStreams Method

Sends the activity to all applicable streams based upon the activity and users preferences.

C#

```
public void SendActivityToStreams(long activityId);
```

Visual Basic

```
Public Function SendActivityToStreams(activityId As long) As void
```

Parameters

Parameters	Description
long activityId	Id of activity to send to streams.

ActivityStream.SendActivityToUsersStream Method

Adds an activity existing in the general feed to a user's feed.

C#

```
public void SendActivityToUsersStream(long activityId, long userId);
```

Visual Basic

```
Public Function SendActivityToUsersStream(activityId As long, userId As long) As void
```

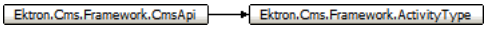
Parameters

Parameters	Description
long activityId	Id of activity to add to users feed.
userid	Id of user.

ActivityType Class

Api class for managing Ektron.Cms (see page 7).

Class Hierarchy



C#

```
public class ActivityType : CmsApi;
```

Visual Basic

```
Public Class ActivityType
Inherits CmsApi
```

File

ActivityType.cs

ActivityType Methods

The methods of the ActivityType class are listed here.

Public Methods

	Name	Description
	Add (see page 1004)	Adds a new ActivityType (see page 1004) to Cms (see page 7). The ActivityType.Id property will be populated with the new ActivityType (see page 1004)'s Id.
	Delete (see page 1004)	Deletes a ActivityType (see page 1004) from the Cms (see page 7).
	GetItem (see page 1005)	Retrieves a ActivityType (see page 1004) by Id.
	GetList (see page 1005)	Retrieves a list.
	Update (see page 1005)	Updates an existing ActivityType (see page 1004) in Cms (see page 7).

ActivityType.Add Method

Adds a new ActivityType (see page 1004) to Cms (see page 7). The ActivityType.Id property will be populated with the new ActivityType (see page 1004)'s Id.

C#

```
public void Add(ActivityTypeData ActivityTypeData);
```

Visual Basic

```
Public Function Add(ActivityTypeData As ActivityTypeData) As void
```

Parameters

Parameters	Description
ActivityTypeData ActivityTypeData	ActivityTypeData object to add.

ActivityType.Delete Method

Deletes a ActivityType (see page 1004) from the Cms (see page 7).

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	Id of ActivityType (see page 1004) to delete.

ActivityType.GetItem Method

Retrieves a ActivityType (see page 1004) by Id.

C#

```
public ActivityTypeData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As ActivityTypeData
```

Parameters

Parameters	Description
long id	Id of ActivityType (see page 1004) to retrieve.

ActivityType.GetList Method

Retrieves a list.

C#

```
public List<ActivityTypeData> GetList(Criteria<ActivityTypeProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<ActivityTypeProperty>) As List<ActivityTypeData>
```

Parameters

Parameters	Description
Criteria<ActivityTypeProperty> criteria	Criteria by which to filter ActivityType (see page 1004) being retrieved.

Returns

List of ActivityType (see page 1004) data objects.

ActivityType.Update Method

Updates an existing ActivityType (see page 1004) in Cms (see page 7).

C#

```
public void Update(ActivityTypeData ActivityTypeData);
```

Visual Basic

```
Public Function Update(ActivityTypeData As ActivityTypeData) As void
```

Parameters

Parameters	Description
ActivityTypeData ActivityTypeData	ActivityTypeData object to save.

Ektron.Cms.Framework.Analytics Namespace

This is namespace Ektron.Cms.Framework.Analytics.

Classes

	Name	Description
	ReportAnalytics (see page 1006)	This is class Ektron.Cms.Framework.Analytics.ReportAnalytics.
	SearchAnalytics (see page 1036)	This is class Ektron.Cms.Framework.Analytics.SearchAnalytics.

Classes

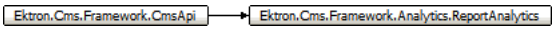
The following table lists classes in this documentation.

Classes

	Name	Description
	ReportAnalytics (see page 1006)	This is class Ektron.Cms.Framework.Analytics.ReportAnalytics.
	SearchAnalytics (see page 1036)	This is class Ektron.Cms.Framework.Analytics.SearchAnalytics.

ReportAnalytics Class

Class Hierarchy



C#

```
public class ReportAnalytics : CmsApi;
```

Visual Basic

```
Public Class ReportAnalytics
Inherits CmsApi
```

File

ReportAnalyticsApi.cs

Description

This is class Ektron.Cms.Framework.Analytics.ReportAnalytics.

ReportAnalytics.ReportAnalytics Constructor

C#

```
public ReportAnalytics(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

ReportAnalytics Methods

The methods of the ReportAnalytics class are listed here.

Public Methods

	Name	Description
	GetAbsoluteUniqueVisitorsTrend (see page 1007)	This is the overview for the GetAbsoluteUniqueVisitorsTrend method overload.
	GetAdVersions (see page 1009)	This is the overview for the GetAdVersions method overload.
	GetAveragePageViewsTrend (see page 1010)	This is the overview for the GetAveragePageViewsTrend method overload.
	GetBounceRateTrend (see page 1011)	This is the overview for the GetBounceRateTrend method overload.
	GetBrowsers (see page 1012)	This is the overview for the GetBrowsers method overload.
	GetCampaigns (see page 1013)	This is the overview for the GetCampaigns method overload.
	GetCmsSearchTerms (see page 1014)	a Cms (see page 7) Search (see page 899) terms report
	GetConnectionSpeeds (see page 1014)	This is the overview for the GetConnectionSpeeds method overload.
	GetContentbyTitle (see page 1015)	This is the overview for the GetContentbyTitle method overload.
	GetContentDetail (see page 1016)	This is the overview for the GetContentDetail method overload.
	GetFlashVersions (see page 1017)	This is the overview for the GetFlashVersions method overload.
	GetHostnames (see page 1018)	This is the overview for the GetHostnames method overload.
	GetJavaSupport (see page 1020)	This is the overview for the GetJavaSupport method overload.
	GetKeywords (see page 1021)	This is the overview for the GetKeywords method overload.
	GetLanguages (see page 1022)	This is the overview for the GetLanguages method overload.
	GetLocations (see page 1023)	This is the overview for the GetLocations method overload.
	GetNetworkLocations (see page 1024)	This is the overview for the GetNetworkLocations method overload.
	GetNewVsReturningVisitors (see page 1025)	This is the overview for the GetNewVsReturningVisitors method overload.
	GetOperatingSystems (see page 1026)	This is the overview for the GetOperatingSystems method overload.
	GetPageViewsTrend (see page 1027)	This is the overview for the GetPageViewsTrend method overload.
	GetPlatforms (see page 1028)	This is the overview for the GetPlatforms method overload.
	GetScreenColors (see page 1029)	This is the overview for the GetScreenColors method overload.
	GetScreenResolutions (see page 1030)	This is the overview for the GetScreenResolutions method overload.
	GetTimeOnSiteTrend (see page 1031)	This is the overview for the GetTimeOnSiteTrend method overload.
	GetTopContent (see page 1032)	This is the overview for the GetTopContent method overload.
	GetTopExitPages (see page 1033)	This is the overview for the GetTopExitPages method overload.
	GetTopLandingPages (see page 1034)	This is the overview for the GetTopLandingPages method overload.
	GetVisitsTrend (see page 1035)	This is the overview for the GetVisitsTrend method overload.

GetAbsoluteUniqueVisitorsTrend Method

This is the overview for the GetAbsoluteUniqueVisitorsTrend method overload.

Overload List

	Name	Description
	ReportAnalytics.GetAbsoluteUniqueVisitorsTrend (DateTime, DateTime, AnalyticsCriteria) (see page 1008)	an absolute unique visitors trend report by the default provider
	ReportAnalytics.GetAbsoluteUniqueVisitorsTrend (string, DateTime, DateTime, AnalyticsCriteria) (see page 1008)	an absolute unique visitors trend report by the specified provider

ReportAnalytics.GetAbsoluteUniqueVisitorsTrend Method (DateTime, DateTime, AnalyticsCriteria)

an absolute unique visitors trend report by the default provider

C#

```
public TrendReportData GetAbsoluteUniqueVisitorsTrend(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetAbsoluteUniqueVisitorsTrend(startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

ReportAnalytics.GetAbsoluteUniqueVisitorsTrend Method (string, DateTime, DateTime, AnalyticsCriteria)

an absolute unique visitors trend report by the specified provider

C#

```
public TrendReportData GetAbsoluteUniqueVisitorsTrend(string provider, DateTime startDate,
DateTime endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetAbsoluteUniqueVisitorsTrend(provider As string, startDate As DateTime,
endDate As DateTime, criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

GetAdVersions Method

This is the overview for the GetAdVersions method overload.

Overload List

	Name	Description
	ReportAnalytics.GetAdVersions (DateTime, DateTime, AnalyticsCriteria) (see page 1009)	an ad versions report by the default provider
	ReportAnalytics.GetAdVersions (string, DateTime, DateTime, AnalyticsCriteria) (see page 1009)	an ad versions report by the specified provider

ReportAnalytics.GetAdVersions Method (DateTime, DateTime, AnalyticsCriteria)

an ad versions report by the default provider

C#

```
public AnalyticsReportData GetAdVersions(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetAdVersions(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetAdVersions Method (string, DateTime, DateTime, AnalyticsCriteria)

an ad versions report by the specified provider

C#

```
public AnalyticsReportData GetAdVersions(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetAdVersions(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetAveragePageViewsTrend Method

This is the overview for the GetAveragePageViewsTrend method overload.

Overload List

	Name	Description
	ReportAnalytics.GetAveragePageViewsTrend (DateTime, DateTime, AnalyticsCriteria) (see page 1010)	an average pageviews for all visitors trend report by the default provider
	ReportAnalytics.GetAveragePageViewsTrend (string, DateTime, DateTime, AnalyticsCriteria) (see page 1010)	an average pageviews for all visitors trend report by the specified provider

ReportAnalytics.GetAveragePageViewsTrend Method (DateTime, DateTime, AnalyticsCriteria)

an average pageviews for all visitors trend report by the default provider

C#

```
public TrendReportData GetAveragePageViewsTrend(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetAveragePageViewsTrend(startDate As DateTime, endDate As DateTime, criteria
As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

ReportAnalytics.GetAveragePageViewsTrend Method (string, DateTime, DateTime, AnalyticsCriteria)

an average pageviews for all visitors trend report by the specified provider

C#

```
public TrendReportData GetAveragePageViewsTrend(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetAveragePageViewsTrend(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report

AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

GetBounceRateTrend Method

This is the overview for the GetBounceRateTrend method overload.

Overload List

	Name	Description
	ReportAnalytics.GetBounceRateTrend (DateTime, DateTime, AnalyticsCriteria) (see page 1011)	a bounce rate for all visitors trend report by the default provider
	ReportAnalytics.GetBounceRateTrend (string, DateTime, DateTime, AnalyticsCriteria) (see page 1011)	a bounce rate for all visitors trend report by the specified provider

ReportAnalytics.GetBounceRateTrend Method (DateTime, DateTime, AnalyticsCriteria)

a bounce rate for all visitors trend report by the default provider

C#

```
public TrendReportData GetBounceRateTrend(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetBounceRateTrend(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

ReportAnalytics.GetBounceRateTrend Method (string, DateTime, DateTime, AnalyticsCriteria)

a bounce rate for all visitors trend report by the specified provider

C#

```
public TrendReportData GetBounceRateTrend(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetBounceRateTrend(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report

DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

GetBrowsers Method

This is the overview for the GetBrowsers method overload.

Overload List

	Name	Description
	ReportAnalytics.GetBrowsers (DateTime, DateTime, AnalyticsCriteria) (see page 1012)	a browser report by the default provider
	ReportAnalytics.GetBrowsers (string, DateTime, DateTime, AnalyticsCriteria) (see page 1012)	a browser report by the specified provider

ReportAnalytics.GetBrowsers Method (DateTime, DateTime, AnalyticsCriteria)

a browser report by the default provider

C#

```
public AnalyticsReportData GetBrowsers(DateTime startDate, DateTime endDate, AnalyticsCriteria
criteria);
```

Visual Basic

```
Public Function GetBrowsers(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetBrowsers Method (string, DateTime, DateTime, AnalyticsCriteria)

a browser report by the specified provider

C#

```
public AnalyticsReportData GetBrowsers(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetBrowsers(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetCampaigns Method

This is the overview for the GetCampaigns method overload.

Overload List

	Name	Description
	ReportAnalytics.GetCampaigns (DateTime, DateTime, AnalyticsCriteria) (see page 1013)	a campaigns report by the default provider
	ReportAnalytics.GetCampaigns (string, DateTime, DateTime, AnalyticsCriteria) (see page 1013)	a campaigns report by the specified provider

ReportAnalytics.GetCampaigns Method (DateTime, DateTime, AnalyticsCriteria)

a campaigns report by the default provider

C#

```
public AnalyticsReportData GetCampaigns(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetCampaigns(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetCampaigns Method (string, DateTime, DateTime, AnalyticsCriteria)

a campaigns report by the specified provider

C#

```
public AnalyticsReportData GetCampaigns(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetCampaigns(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetCmsSearchTerms Method

a Cms (see page 7) Search (see page 899) terms report

C#

```
public AnalyticsReportData GetCmsSearchTerms(DateTime startDate, DateTime endDate);
```

Visual Basic

```
Public Function GetCmsSearchTerms(startDate As DateTime, endDate As DateTime) As
AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetConnectionSpeeds Method

This is the overview for the GetConnectionSpeeds method overload.

Overload List

	Name	Description
	ReportAnalytics.GetConnectionSpeeds (DateTime, DateTime, AnalyticsCriteria) (see page 1014)	an user defined value report by the specified provider
	ReportAnalytics.GetConnectionSpeeds (string, DateTime, DateTime, AnalyticsCriteria) (see page 1015)	a connection speeds report by the specified provider

ReportAnalytics.GetConnectionSpeeds Method (DateTime, DateTime, AnalyticsCriteria)

an user defined value report by the specified provider

C#

```
public AnalyticsReportData GetConnectionSpeeds(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetConnectionSpeeds(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query
provider	the provider to carry out the report

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetConnectionSpeeds Method (string, DateTime, DateTime, AnalyticsCriteria)

a connection speeds report by the specified provider

C#

```
public AnalyticsReportData GetConnectionSpeeds(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetConnectionSpeeds(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetContentbyTitle Method

This is the overview for the GetContentbyTitle method overload.

Overload List

	Name	Description
	ReportAnalytics.GetContentbyTitle (DateTime, DateTime, AnalyticsCriteria) (see page 1015)	a content by title report by the default provider
	ReportAnalytics.GetContentbyTitle (string, DateTime, DateTime, AnalyticsCriteria) (see page 1016)	a content by title report by the specified provider

ReportAnalytics.GetContentbyTitle Method (DateTime, DateTime, AnalyticsCriteria)

a content by title report by the default provider

C#

```
public AnalyticsReportData GetContentbyTitle(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetContentbyTitle(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetContentbyTitle Method (string, DateTime, DateTime, AnalyticsCriteria)

a content by title report by the specified provider

C#

```
public AnalyticsReportData GetContentbyTitle(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetContentbyTitle(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetContentDetail Method

This is the overview for the GetContentDetail method overload.

Overload List

	Name	Description
	ReportAnalytics.GetContentDetail (DateTime, DateTime, AnalyticsCriteria) (see page 1017)	a top exit pages report by the specified provider
	ReportAnalytics.GetContentDetail (string, DateTime, DateTime, AnalyticsCriteria) (see page 1017)	a top exit pages report by the specified provider

ReportAnalytics.GetContentDetail Method (DateTime, DateTime, AnalyticsCriteria)

a top exit pages report by the specified provider

C#

```
public AnalyticsReportData GetContentDetail(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetContentDetail(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetContentDetail Method (string, DateTime, DateTime, AnalyticsCriteria)

a top exit pages report by the specified provider

C#

```
public AnalyticsReportData GetContentDetail(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetContentDetail(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetFlashVersions Method

This is the overview for the GetFlashVersions method overload.

Overload List

	Name	Description
	ReportAnalytics.GetFlashVersions (DateTime, DateTime, AnalyticsCriteria) (see page 1018)	a flash versions report by the default provider

	ReportAnalytics.GetFlashVersions (string, DateTime, DateTime, AnalyticsCriteria) (see page 1018)	a flash versions report by the specified provider

ReportAnalytics.GetFlashVersions Method (DateTime, DateTime, AnalyticsCriteria)

a flash versions report by the default provider

C#

```
public AnalyticsReportData GetFlashVersions(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetFlashVersions(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetFlashVersions Method (string, DateTime, DateTime, AnalyticsCriteria)

a flash versions report by the specified provider

C#

```
public AnalyticsReportData GetFlashVersions(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetFlashVersions(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetHostnames Method

This is the overview for the GetHostnames method overload.

Overload List

	Name	Description
	ReportAnalytics.GetHostnames (DateTime, DateTime, AnalyticsCriteria) (see page 1019)	a hostnames report by the default provider
	ReportAnalytics.GetHostnames (string, DateTime, DateTime, AnalyticsCriteria) (see page 1019)	a hostnames report by the specified provider

ReportAnalytics.GetHostnames Method (DateTime, DateTime, AnalyticsCriteria)

a hostnames report by the default provider

C#

```
public AnalyticsReportData GetHostnames(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetHostnames(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetHostnames Method (string, DateTime, DateTime, AnalyticsCriteria)

a hostnames report by the specified provider

C#

```
public AnalyticsReportData GetHostnames(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetHostnames(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetJavaSupport Method

This is the overview for the GetJavaSupport method overload.

Overload List

	Name	Description
	ReportAnalytics.GetJavaSupport (DateTime, DateTime, AnalyticsCriteria) (see page 1020)	a java support report by the default provider
	ReportAnalytics.GetJavaSupport (string, DateTime, DateTime, AnalyticsCriteria) (see page 1020)	a java support report by the specified provider

ReportAnalytics.GetJavaSupport Method (DateTime, DateTime, AnalyticsCriteria)

a java support report by the default provider

C#

```
public AnalyticsReportData GetJavaSupport(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetJavaSupport(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetJavaSupport Method (string, DateTime, DateTime, AnalyticsCriteria)

a java support report by the specified provider

C#

```
public AnalyticsReportData GetJavaSupport(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetJavaSupport(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetKeywords Method

This is the overview for the GetKeywords method overload.

Overload List

	Name	Description
	ReportAnalytics.GetKeywords (DateTime, DateTime, AnalyticsCriteria) (see page 1021)	a keywords report by the default provider
	ReportAnalytics.GetKeywords (string, DateTime, DateTime, AnalyticsCriteria) (see page 1021)	a keywords report by the specified provider

ReportAnalytics.GetKeywords Method (DateTime, DateTime, AnalyticsCriteria)

a keywords report by the default provider

C#

```
public AnalyticsReportData GetKeywords(DateTime startDate, DateTime endDate, AnalyticsCriteria
criteria);
```

Visual Basic

```
Public Function GetKeywords(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetKeywords Method (string, DateTime, DateTime, AnalyticsCriteria)

a keywords report by the specified provider

C#

```
public AnalyticsReportData GetKeywords(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetKeywords(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report

DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetLanguages Method

This is the overview for the GetLanguages method overload.

Overload List

	Name	Description
	ReportAnalytics.GetLanguages (DateTime, DateTime, AnalyticsCriteria) (see page 1022)	a report on different languages used by the visitor by the default provider
	ReportAnalytics.GetLanguages (string, DateTime, DateTime, AnalyticsCriteria) (see page 1022)	a report on different languages used by the visitor by the specified provider

ReportAnalytics.GetLanguages Method (DateTime, DateTime, AnalyticsCriteria)

a report on different languages used by the visitor by the default provider

C#

```
public AnalyticsReportData GetLanguages(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetLanguages(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetLanguages Method (string, DateTime, DateTime, AnalyticsCriteria)

a report on different languages used by the visitor by the specified provider

C#

```
public AnalyticsReportData GetLanguages(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetLanguages(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetLocations Method

This is the overview for the GetLocations method overload.

Overload List

	Name	Description
	ReportAnalytics.GetLocations (DateTime, DateTime, AnalyticsCriteria) (see page 1023)	an user geographic location report by the default provider
	ReportAnalytics.GetLocations (string, DateTime, DateTime, AnalyticsCriteria) (see page 1023)	an user geographic location report by the specified provider

ReportAnalytics.GetLocations Method (DateTime, DateTime, AnalyticsCriteria)

an user geographic location report by the default provider

C#

```
public AnalyticsReportData GetLocations(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetLocations(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetLocations Method (string, DateTime, DateTime, AnalyticsCriteria)

an user geographic location report by the specified provider

C#

```
public AnalyticsReportData GetLocations(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetLocations(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetNetworkLocations Method

This is the overview for the GetNetworkLocations method overload.

Overload List

	Name	Description
	ReportAnalytics.GetNetworkLocations (DateTime, DateTime, AnalyticsCriteria) (see page 1024)	a network location report by the default provider
	ReportAnalytics.GetNetworkLocations (string, DateTime, DateTime, AnalyticsCriteria) (see page 1024)	a network location report by the specified provider

ReportAnalytics.GetNetworkLocations Method (DateTime, DateTime, AnalyticsCriteria)

a network location report by the default provider

C#

```
public AnalyticsReportData GetNetworkLocations(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetNetworkLocations(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetNetworkLocations Method (string, DateTime, DateTime, AnalyticsCriteria)

a network location report by the specified provider

C#

```
public AnalyticsReportData GetNetworkLocations(string provider, DateTime startDate, DateTime
```

```
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetNetworkLocations(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetNewVsReturningVisitors Method

This is the overview for the GetNewVsReturningVisitors method overload.

Overload List

	Name	Description
	ReportAnalytics.GetNewVsReturningVisitors (DateTime, DateTime, AnalyticsCriteria) (see page 1025)	a report on new visitor vs. returning visitor by the default provider
	ReportAnalytics.GetNewVsReturningVisitors (string, DateTime, DateTime, AnalyticsCriteria) (see page 1025)	a report on new visitor vs. returning visitor by the specified provider

ReportAnalytics.GetNewVsReturningVisitors Method (DateTime, DateTime, AnalyticsCriteria)

a report on new visitor vs. returning visitor by the default provider

C#

```
public AnalyticsReportData GetNewVsReturningVisitors(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetNewVsReturningVisitors(startDate As DateTime, endDate As DateTime, criteria
As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetNewVsReturningVisitors Method (string, DateTime, DateTime, AnalyticsCriteria)

a report on new visitor vs. returning visitor by the specified provider

C#

```
public AnalyticsReportData GetNewVsReturningVisitors(string provider, DateTime startDate,
DateTime endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetNewVsReturningVisitors(provider As string, startDate As DateTime, endDate
As DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetOperatingSystems Method

This is the overview for the GetOperatingSystems method overload.

Overload List

	Name	Description
	ReportAnalytics.GetOperatingSystems (DateTime, DateTime, AnalyticsCriteria) (see page 1026)	an operating systems report by the default provider
	ReportAnalytics.GetOperatingSystems (string, DateTime, DateTime, AnalyticsCriteria) (see page 1027)	an operating systems report by the specified provider

ReportAnalytics.GetOperatingSystems Method (DateTime, DateTime, AnalyticsCriteria)

an operating systems report by the default provider

C#

```
public AnalyticsReportData GetOperatingSystems(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetOperatingSystems(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetOperatingSystems Method (string, DateTime, DateTime, AnalyticsCriteria)

an operating systems report by the specified provider

C#

```
public AnalyticsReportData GetOperatingSystems(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetOperatingSystems(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetPageViewsTrend Method

This is the overview for the GetPageViewsTrend method overload.

Overload List

	Name	Description
	ReportAnalytics.GetPageViewsTrend (DateTime, DateTime, AnalyticsCriteria) (see page 1027)	a pageviews for all visitors trend report by the default provider
	ReportAnalytics.GetPageViewsTrend (string, DateTime, DateTime, AnalyticsCriteria) (see page 1028)	a pageviews for all visitors trend report by the specified provider

ReportAnalytics.GetPageViewsTrend Method (DateTime, DateTime, AnalyticsCriteria)

a pageviews for all visitors trend report by the default provider

C#

```
public TrendReportData GetPageViewsTrend(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetPageViewsTrend(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

ReportAnalytics.GetPageViewsTrend Method (string, DateTime, DateTime, AnalyticsCriteria)

a pageviews for all visitors trend report by the specified provider

C#

```
public TrendReportData GetPageViewsTrend(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetPageViewsTrend(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

GetPlatforms Method

This is the overview for the GetPlatforms method overload.

Overload List

	Name	Description
	ReportAnalytics.GetPlatforms (DateTime, DateTime, AnalyticsCriteria) (see page 1028)	a platforms report by the default provider
	ReportAnalytics.GetPlatforms (string, DateTime, DateTime, AnalyticsCriteria) (see page 1029)	a platforms report by the specified provider

ReportAnalytics.GetPlatforms Method (DateTime, DateTime, AnalyticsCriteria)

a platforms report by the default provider

C#

```
public AnalyticsReportData GetPlatforms(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetPlatforms(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetPlatforms Method (string, DateTime, DateTime, AnalyticsCriteria)

a platforms report by the specified provider

C#

```
public AnalyticsReportData GetPlatforms(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetPlatforms(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetScreenColors Method

This is the overview for the GetScreenColors method overload.

Overload List

	Name	Description
	ReportAnalytics.GetScreenColors (DateTime, DateTime, AnalyticsCriteria) (see page 1029)	a screen colors report by the default provider
	ReportAnalytics.GetScreenColors (string, DateTime, DateTime, AnalyticsCriteria) (see page 1030)	a screen colors report by the specified provider

ReportAnalytics.GetScreenColors Method (DateTime, DateTime, AnalyticsCriteria)

a screen colors report by the default provider

C#

```
public AnalyticsReportData GetScreenColors(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetScreenColors(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetScreenColors Method (string, DateTime, DateTime, AnalyticsCriteria)

a screen colors report by the specified provider

C#

```
public AnalyticsReportData GetScreenColors(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetScreenColors(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetScreenResolutions Method

This is the overview for the GetScreenResolutions method overload.

Overload List

	Name	Description
	ReportAnalytics.GetScreenResolutions (DateTime, DateTime, AnalyticsCriteria) (see page 1030)	a screen resolutions report by the default provider
	ReportAnalytics.GetScreenResolutions (string, DateTime, DateTime, AnalyticsCriteria) (see page 1031)	a screen resolutions report by the specified provider

ReportAnalytics.GetScreenResolutions Method (DateTime, DateTime, AnalyticsCriteria)

a screen resolutions report by the default provider

C#

```
public AnalyticsReportData GetScreenResolutions(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetScreenResolutions(startDate As DateTime, endDate As DateTime, criteria As
```

```
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetScreenResolutions Method (string, DateTime, DateTime, AnalyticsCriteria)

a screen resolutions report by the specified provider

C#

```
public AnalyticsReportData GetScreenResolutions(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetScreenResolutions(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetTimeOnSiteTrend Method

This is the overview for the GetTimeOnSiteTrend method overload.

Overload List

	Name	Description
	ReportAnalytics.GetTimeOnSiteTrend (DateTime, DateTime, AnalyticsCriteria) (see page 1031)	a time on site for all visitors trend report by the default provider
	ReportAnalytics.GetTimeOnSiteTrend (string, DateTime, DateTime, AnalyticsCriteria) (see page 1032)	a time on site for all visitors trend report by the specified provider

ReportAnalytics.GetTimeOnSiteTrend Method (DateTime, DateTime, AnalyticsCriteria)

a time on site for all visitors trend report by the default provider

C#

```
public TrendReportData GetTimeOnSiteTrend(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetTimeOnSiteTrend(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

ReportAnalytics.GetTimeOnSiteTrend Method (string, DateTime, DateTime, AnalyticsCriteria)

a time on site for all visitors trend report by the specified provider

C#

```
public TrendReportData GetTimeOnSiteTrend(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetTimeOnSiteTrend(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

GetTopContent Method

This is the overview for the GetTopContent method overload.

Overload List

	Name	Description
	ReportAnalytics.GetTopContent (DateTime, DateTime, AnalyticsCriteria) (see page 1032)	a top content report by the default provider
	ReportAnalytics.GetTopContent (string, DateTime, DateTime, AnalyticsCriteria) (see page 1033)	a top content report by the specified provider

ReportAnalytics.GetTopContent Method (DateTime, DateTime, AnalyticsCriteria)

a top content report by the default provider

C#

```
public AnalyticsReportData GetTopContent(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetTopContent(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetTopContent Method (string, DateTime, DateTime, AnalyticsCriteria)

a top content report by the specified provider

C#

```
public AnalyticsReportData GetTopContent(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetTopContent(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetTopExitPages Method

This is the overview for the GetTopExitPages method overload.

Overload List

	Name	Description
	ReportAnalytics.GetTopExitPages (DateTime, DateTime, AnalyticsCriteria) (see page 1034)	a top exit pages report by the default provider
	ReportAnalytics.GetTopExitPages (string, DateTime, DateTime, AnalyticsCriteria) (see page 1034)	a top exit pages report by the specified provider

ReportAnalytics.GetTopExitPages Method (DateTime, DateTime, AnalyticsCriteria)

a top exit pages report by the default provider

C#

```
public AnalyticsReportData GetTopExitPages(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetTopExitPages(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetTopExitPages Method (string, DateTime, DateTime, AnalyticsCriteria)

a top exit pages report by the specified provider

C#

```
public AnalyticsReportData GetTopExitPages(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetTopExitPages(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetTopLandingPages Method

This is the overview for the GetTopLandingPages method overload.

Overload List

	Name	Description
	ReportAnalytics.GetTopLandingPages (DateTime, DateTime, AnalyticsCriteria) (see page 1035)	a top landing pages report by the default provider
	ReportAnalytics.GetTopLandingPages (string, DateTime, DateTime, AnalyticsCriteria) (see page 1035)	a top landing pages report by the specified provider

ReportAnalytics.GetTopLandingPages Method (DateTime, DateTime, AnalyticsCriteria)

a top landing pages report by the default provider

C#

```
public AnalyticsReportData GetTopLandingPages(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetTopLandingPages(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

ReportAnalytics.GetTopLandingPages Method (string, DateTime, DateTime, AnalyticsCriteria)

a top landing pages report by the specified provider

C#

```
public AnalyticsReportData GetTopLandingPages(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetTopLandingPages(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetVisitsTrend Method

This is the overview for the GetVisitsTrend method overload.

Overload List

	Name	Description
	ReportAnalytics.GetVisitsTrend (DateTime, DateTime, AnalyticsCriteria) (see page 1036)	a visitors trend report by the default provider

	ReportAnalytics.GetVisitsTrend (string, DateTime, DateTime, AnalyticsCriteria) (see page 1036)	a visitors trend report by the specified provider

ReportAnalytics.GetVisitsTrend Method (DateTime, DateTime, AnalyticsCriteria)

a visitors trend report by the default provider

C#

```
public TrendReportData GetVisitsTrend(DateTime startDate, DateTime endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetVisitsTrend(startDate As DateTime, endDate As DateTime, criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

ReportAnalytics.GetVisitsTrend Method (string, DateTime, DateTime, AnalyticsCriteria)

a visitors trend report by the specified provider

C#

```
public TrendReportData GetVisitsTrend(string provider, DateTime startDate, DateTime endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
Public Function GetVisitsTrend(provider As string, startDate As DateTime, endDate As DateTime, criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

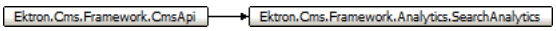
Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

SearchAnalytics Class

Class Hierarchy



C#

```
public class SearchAnalytics : CmsApi;
```

Visual Basic

```
Public Class SearchAnalytics
Inherits CmsApi
```

File

SearchAnalyticsApi.cs

Description

This is class Ektron.Cms.Framework.Analytics.SearchAnalytics.

SearchAnalytics.SearchAnalytics Constructor

C#

```
public SearchAnalytics(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

Ektron.Cms.Framework.Calendar Namespace

This is namespace Ektron.Cms.Framework.Calendar.

Classes

	Name	Description
	WebCalendar (see page 1037)	This is class Ektron.Cms.Framework.Calendar.WebCalendar.
	WebEvent (see page 1040)	This is class Ektron.Cms.Framework.Calendar.WebEvent.

Classes

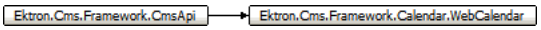
The following table lists classes in this documentation.

Classes

	Name	Description
	WebCalendar (see page 1037)	This is class Ektron.Cms.Framework.Calendar.WebCalendar.
	WebEvent (see page 1040)	This is class Ektron.Cms.Framework.Calendar.WebEvent.

WebCalendar Class

Class Hierarchy



C#

```
public class WebCalendar : CmsApi;
```

Visual Basic

```
Public Class WebCalendar
Inherits CmsApi
```

File

WebCalendar.cs

Description

This is class Ektron.Cms.Framework.Calendar.WebCalendar.

WebCalendar Constructor

This is the overview for the WebCalendar constructor overload.

Overload List

	Name	Description
	WebCalendar.WebCalendar () (see page 1038)	constructor
	WebCalendar.WebCalendar (ApiAccessMode) (see page 1038)	constructor

WebCalendar.WebCalendar Constructor ()

constructor

C#

```
public WebCalendar();
```

Visual Basic

```
Public Sub New()
```

WebCalendar.WebCalendar Constructor (ApiAccessMode)

constructor

C#

```
public WebCalendar(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

WebCalendar Methods

The methods of the WebCalendar class are listed here.

Public Methods

	Name	Description
	Add (see page 1039)	Saves the current calendar as a new calendar.
	Delete (see page 1039)	Deletes the current calendar.
	GetCalendar (see page 1039)	Retrieves a calendar.
	GetICalendarFeed (see page 1039)	Returns the ICalendar representation of the supplied calendar Id.

	GetPublicCalendar (see page 1039)	Retrieves a calendar with only public fields populated.
	Update (see page 1040)	Saves this calendar.

WebCalendar.Add Method

Saves the current calendar as a new calendar.

C#

```
public void Add(WebCalendarData calendar);
```

Visual Basic

```
Public Function Add(calendar As WebCalendarData) As void
```

WebCalendar.Delete Method

Deletes the current calendar.

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

WebCalendar.GetCalendar Method

Retrieves a calendar.

C#

```
public WebCalendarData GetCalendar(long calendarId);
```

Visual Basic

```
Public Function GetCalendar(calendarId As long) As WebCalendarData
```

Parameters

Parameters	Description
long calendarId	Id of the calendar to retrieve.

Returns

WebCalendarData

WebCalendar.GetICalendarFeed Method

Returns the ICalendar representation of the supplied calendar Id.

C#

```
public string GetICalendarFeed(long calendarId);
```

Visual Basic

```
Public Function GetICalendarFeed(calendarId As long) As string
```

Parameters

Parameters	Description
calendar	Id of calendar to retrieve ICalendar feed for.

WebCalendar.GetPublicCalendar Method

Retrieves a calendar with only public fields populated.

C#

```
public WebCalendarData GetPublicCalendar(long calendarId);
```

Visual Basic

```
Public Function GetPublicCalendar(calendarId As long) As WebCalendarData
```

Parameters

Parameters	Description
long calendarId	Id of the calendar to retrieve.

Returns

WebCalendarData with only public fields populated.

WebCalendar.Update Method

Saves this calendar.

C#

```
public void Update(WebCalendarData calendar);
```

Visual Basic

```
Public Function Update(calendar As WebCalendarData) As void
```

WebEvent Class

Class Hierarchy

C#

```
public class WebEvent : CmsApi;
```

Visual Basic

```
Public Class WebEvent
Inherits CmsApi
```

File

WebEvent.cs

Description

This is class Ektron.Cms.Framework.Calendar.WebEvent.

WebEvent Constructor

This is the overview for the WebEvent constructor overload.

Overload List

	Name	Description
	WebEvent.WebEvent () (see page 1041)	constructor
	WebEvent.WebEvent (ApiAccessMode) (see page 1041)	constructor

WebEvent.WebEvent Constructor ()

constructor

C#

```
public WebEvent();
```

Visual Basic

```
Public Sub New()
```

WebEvent.WebEvent Constructor (ApiAccessMode)

constructor

C#

```
public WebEvent(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

WebEvent Methods

The methods of the WebEvent class are listed here.

Public Methods

	Name	Description
	Add (see page 1041)	Adds a new WebEvent (see page 1040) object to Cms (see page 7). The WebEvent.Id property will be populated with the new WebEvent (see page 1040)'s Id.
	CancelOccurrence (see page 1042)	Cancels an occurence of a recurring event.
	CreateVariance (see page 1042)	This is the overview for the CreateVariance method overload.
	Delete (see page 1043)	Deletes a WebEvent (see page 1040) from the Cms (see page 7).
	GetEventOccurrenceList (see page 1043)	Returns a list of events for the calendar that occur during the supplied time period. This will return all event occurrences including individual occurrences for recurring events.
	GetItem (see page 1044)	Gets the fully populated EventData object for the supplied eventId.
	GetList (see page 1044)	This is the overview for the GetList method overload.
	GetNonVariantEventList (see page 1045)	This is the overview for the GetNonVariantEventList method overload.
	GetVarianceEventList (see page 1046)	This is the overview for the GetVarianceEventList method overload.
	Update (see page 1047)	Updates an existing WebEventData object in Cms (see page 7).

WebEvent.Add Method

Adds a new WebEvent (see page 1040) object to Cms (see page 7). The WebEvent.Id property will be populated with the new WebEvent (see page 1040)'s Id.

C#

```
public void Add(WebEventData eventData);
```

Visual Basic

```
Public Function Add(eventData As WebEventData) As void
```

Parameters

Parameters	Description
WebEventData eventData	WebEventData obejct to add.

WebEvent.CancelOccurrence Method

Cancels an occurence of a recurring event.

C#

```
public long CancelOccurrence(long eventId, DateTime occurrenceStartUtc);
```

Visual Basic

```
Public Function CancelOccurrence(eventId As long, occurrenceStartUtc As DateTime) As long
```

Parameters

Parameters	Description
long eventId	Id of the recurring event to cancel an occurrence for.
DateTime occurrenceStartUtc	The event occurance date to cancel.

Returns

Id of the event variance created for the cancellation.

CreateVariance Method

This is the overview for the CreateVariance method overload.

Overload List

	Name	Description
	WebEvent.CreateVariance (long, DateTime) (see page 1042)	Creates a variance event for the supplied recurring event id.
	WebEvent.CreateVariance (long, DateTime, DateTime) (see page 1043)	Creates a variance event for the supplied recurring event id.

WebEvent.CreateVariance Method (long, DateTime)

Creates a variance event for the supplied recurring event id.

C#

```
public WebEventData CreateVariance(long eventId, DateTime occurrenceStartUtc);
```

Visual Basic

```
Public Function CreateVariance(eventId As long, occurrenceStartUtc As DateTime) As WebEventData
```

Parameters

Parameters	Description
long eventId	Id of the recurring event to create variance for.
DateTime occurrenceStartUtc	The event occurance date for which to create a variance.

Returns

An unsaved WebEvent (see page 1040) representing the event variance/

WebEvent.CreateVariance Method (long, DateTime, DateTime)

Creates a variance event for the supplied recurring event id.

C#

```
public WebEventData CreateVariance(long eventId, DateTime occurrenceStartUtc, DateTime
newStartUtc);
```

Visual Basic

```
Public Function CreateVariance(eventId As long, occurrenceStartUtc As DateTime, newStartUtc As
DateTime) As WebEventData
```

Parameters

Parameters	Description
long eventId	Id of the recurring event to create variance for.
DateTime occurrenceStartUtc	The event occurance date for which to create a variance.
DateTime newStartUtc	The new occureance start date (UTC).

Returns

An unsaved WebEvent (see page 1040) representing the event variance/

WebEvent.Delete Method

Deletes a WebEvent (see page 1040) from the Cms (see page 7).

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	Id of WebEvent (see page 1040) to delete.

WebEvent.GetEventOccurrenceList Method

Returns a list of events for the calendar that occur during the supplied time period. This will return all event occurrences including individual occurrences for recurring events.

C#

```
public List<WebEventData> GetEventOccurrenceList(long calendarId, DateTime startUtc, DateTime
endUtc);
```

Visual Basic

```
Public Function GetEventOccurrenceList(calendarId As long, startUtc As DateTime, endUtc As
DateTime) As List<WebEventData>
```

Parameters

Parameters	Description
long calendarId	Id of the calendar to retrieve events for.
DateTime startUtc	start of search span
DateTime endUtc	end of search span
languageId	Id of of content language to retrieve.

Remarks

EventData objects are not fully populated. Only contains enough data to display event information and retrieve full EventData

object when neccesary.

WebEvent.GetItem Method

Gets the fully populated EventData object for the supplied eventId.

C#

```
public WebEventData GetItem(long eventId);
```

Visual Basic

```
Public Function GetItem(eventId As long) As WebEventData
```

Parameters

Parameters	Description
long eventId	Id of the event to retrieve.

GetList Method

This is the overview for the GetList method overload.

Overload List

	Name	Description
	WebEvent.GetList (Criteria<EventProperty>) (see page 1044)	Returns a list of WebEvents based upon the supplied criteria.
	WebEvent.GetList (long) (see page 1044)	Returns list of eventdata items for the calendar. This will only return the actual event items and not the occurrences of the event.
	WebEvent.GetList (long, DateTime, DateTime) (see page 1045)	Returns list of eventdata items for the calendar. This will only return the actual event items and not the occurrences of the event.

WebEvent.GetList Method (Criteria<EventProperty>)

Returns a list of WebEvents based upon the supplied criteria.

C#

```
public List<WebEventData> GetList(Criteria<EventProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<EventProperty>) As List<WebEventData>
```

Parameters

Parameters	Description
Criteria<EventProperty> criteria	criteria used to restrict WebEvent (see page 1040) retrieval.

WebEvent.GetList Method (long)

Returns list of eventdata items for the calendar. This will only return the actual event items and not the occurrences of the event.

C#

```
public List<WebEventData> GetList(long calendarId);
```

Visual Basic

```
Public Function GetList(calendarId As long) As List<WebEventData>
```

Parameters

Parameters	Description
long calendarId	Id of the calendar to retrieve events for.

Remarks

EventData objects are not fully populated. Only contains enough data to display event information and retrieve full EventData object when neccesary.

WebEvent.GetList Method (long, DateTime, DateTime)

Returns list of eventdata items for the calendar. This will only return the actual event items and not the occurrences of the event.

C#

```
public List<WebEventData> GetList(long calendarId, DateTime startUtc, DateTime endUtc);
```

Visual Basic

```
Public Function GetList(calendarId As long, startUtc As DateTime, endUtc As DateTime) As
List<WebEventData>
```

Parameters

Parameters	Description
long calendarId	Id of the calendar to retrieve events for.
DateTime startUtc	start of time period to retrieve.
DateTime endUtc	end of time period to retrieve.

Remarks

EventData objects are not fully populated. Only contains enough data to display event information and retrieve full EventData object when neccesary.

GetNonVariantEventList Method

This is the overview for the GetNonVariantEventList method overload.

Overload List

	Name	Description
	WebEvent.GetNonVariantEventList (long) (see page 1045)	Returns list of original eventdata items for the calendar. For recurring events, this will only return the actual event items and not the occurrences or variants of recurring events.
	WebEvent.GetNonVariantEventList (long, DateTime, DateTime) (see page 1046)	Returns list of original eventdata items for the calendar. For recurring events, this will only return the actual event items and not the occurrences or variants of recurring events.

WebEvent.GetNonVariantEventList Method (long)

Returns list of original eventdata items for the calendar. For recurring events, this will only return the actual event items and not the occurrences or variants of recurring events.

C#

```
public List<WebEventData> GetNonVariantEventList(long calendarId);
```

Visual Basic

```
Public Function GetNonVariantEventList(calendarId As long) As List<WebEventData>
```

Parameters

Parameters	Description
long calendarId	Id of the calendar to retrieve events for.

WebEvent.GetNonVariantEventList Method (long, DateTime, DateTime)

Returns list of original eventdata items for the calendar. For recurring events, this will only return the actual event items and not the occurrences or variants of recurring events.

C#

```
public List<WebEventData> GetNonVariantEventList(long calendarId, DateTime startUtc, DateTime
endUtc);
```

Visual Basic

```
Public Function GetNonVariantEventList(calendarId As long, startUtc As DateTime, endUtc As
DateTime) As List<WebEventData>
```

Parameters

Parameters	Description
long calendarId	Id of the calendar to retrieve events for.
DateTime startUtc	start of time period to retrieve.
DateTime endUtc	end of time period to retrieve.
filterNonReadableContent	if true, will restrict WebEvent (see page 1040) list to only WebEvents readable by current user in workarea based upon user permisisons.

GetVarianceEventList Method

This is the overview for the GetVarianceEventList method overload.

Overload List

	Name	Description
	WebEvent.GetVarianceEventList (long) (see page 1046)	Returns an EventVarianceDictionary containing all the variances for the supplied claendar in the given time frame.
	WebEvent.GetVarianceEventList (long, DateTime, DateTime) (see page 1046)	Returns an EventVarianceDictionary containing all the variances for the supplied claendar in the given time frame.

WebEvent.GetVarianceEventList Method (long)

Returns an EventVarianceDictionary containing all the variances for the supplied claendar in the given time frame.

C#

```
public WebEventVarianceDictionary GetVarianceEventList(long calendarId);
```

Visual Basic

```
Public Function GetVarianceEventList(calendarId As long) As WebEventVarianceDictionary
```

Parameters

Parameters	Description
long calendarId	Id of the calendar to retrieve events for.

WebEvent.GetVarianceEventList Method (long, DateTime, DateTime)

Returns an EventVarianceDictionary containing all the variances for the supplied claendar in the given time frame.

C#

```
public WebEventVarianceDictionary GetVarianceEventList(long calendarId, DateTime startUtc,
DateTime endUtc);
```

Visual Basic

```
Public Function GetVarianceEventList(calendarId As long, startUtc As DateTime, endUtc As
DateTime) As WebEventVarianceDictionary
```

Parameters

Parameters	Description
long calendarId	Id of the calendar to retrieve events for.
DateTime startUtc	start of time period to retrieve.
DateTime endUtc	end of time period to retrieve.

WebEvent.Update Method

Updates an existing WebEventData object in Cms (see page 7).

C#

```
public void Update(WebEventData eventData);
```

Visual Basic

```
Public Function Update(eventData As WebEventData) As void
```

Parameters

Parameters	Description
WebEventData eventData	WebEventData obejct to save.

Ektron.Cms.Framework.Core Namespace

This is namespace Ektron.Cms.Framework.Core.

Namespaces

Name	Description
Content (see page 1047)	This is namespace Ektron.Cms.Framework.Core.Content.
Folder (see page 1050)	This is namespace Ektron.Cms.Framework.Core.Folder.

Ektron.Cms.Framework.Core.Content Namespace

This is namespace Ektron.Cms.Framework.Core.Content.

Classes

	Name	Description
	Content (see page 1048)	This is class Ektron.Cms.Framework.Core.Content.Content.

Classes

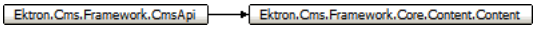
The following table lists classes in this documentation.

Classes

	Name	Description
	Content (see page 1048)	This is class Ektron.Cms.Framework.Core.Content.Content.

Content Class

Class Hierarchy



C#

```
public class Content : CmsApi;
```

Visual Basic

```
Public Class Content
Inherits CmsApi
```

File

Content.cs

Description

This is class Ektron.Cms.Framework.Core.Content.Content.

Content.Content Constructor

C#

```
public Content(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator

Content Methods

The methods of the Content class are listed here.

Public Methods

	Name	Description
	Add (see page 1048)	Add's new content
	Delete (see page 1049)	Delete content
	GetItem (see page 1049)	Get the content item
	GetList (see page 1049)	Retrieves a list based on the criteria
	Update (see page 1049)	Update content

Content.Add Method

Add's new content

C#

```
public void Add(ContentData contentData);
```

Visual Basic

```
Public Function Add(contentData As ContentData) As void
```

Parameters

Parameters	Description
ContentData contentData	The contentData object

Content.Delete Method

Delete content

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	The ID of the content to be deleted

Content.GetItem Method

Get the content item

C#

```
public ContentData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As ContentData
```

Parameters

Parameters	Description
long id	The ID of the content to Get.

Content.GetList Method

Retrieves a list based on the criteria

C#

```
public List<ContentData> GetList(Criteria<ContentProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<ContentProperty>) As List<ContentData>
```

Parameters

Parameters	Description
Criteria<ContentProperty> criteria	Criteria used to retrieve content.

Content.Update Method

Update content

C#

```
public void Update(ContentData contentData);
```

Visual Basic

```
Public Function Update(contentData As ContentData) As void
```

Parameters

Parameters	Description
ContentData contentData	The contentData object

Ektron.Cms.Framework.Core.Folder Namespace

This is namespace Ektron.Cms.Framework.Core.Folder.

Classes

	Name	Description
	Folder (see page 1050)	This is class Ektron.Cms.Framework.Core.Folder.Folder.

Classes

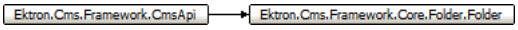
The following table lists classes in this documentation.

Classes

	Name	Description
	Folder (see page 1050)	This is class Ektron.Cms.Framework.Core.Folder.Folder.

Folder Class

Class Hierarchy



C#

```
public class Folder : CmsApi;
```

Visual Basic

```
Public Class Folder
Inherits CmsApi
```

File

Folder.cs

Description

This is class Ektron.Cms.Framework.Core.Folder.Folder.

Folder.Folder Constructor

C#

```
public Folder(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

Folder Methods

The methods of the Folder class are listed here.

Public Methods

	Name	Description
	Add (see page 1051)	Adds a new folder based on information in an FolderData object. message.Id will be populated with the newly created ID.

	Delete (see page 1051)	Deletes a folder in the CMS.
	GetItem (see page 1051)	Retrieves a single FolderData object from the CMS.
	GetList (see page 1051)	Retrieves a list of FolderData objects based upon the supplied criteria.
	Update (see page 1052)	Updates an existing folder in the CMS.

Folder.Add Method

Adds a new folder based on information in an FolderData object. message.Id will be populated with the newly created ID.

C#

```
public void Add(FolderData folderData);
```

Visual Basic

```
Public Function Add(folderData As FolderData) As void
```

Parameters

Parameters	Description
FolderData folderData	The FolderData object to add.

Folder.Delete Method

Deletes a folder in the CMS.

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	The ID of the folder to delete

Folder.GetItem Method

Retrieves a single FolderData object from the CMS.

C#

```
public FolderData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As FolderData
```

Parameters

Parameters	Description
long id	The ID of the folder to retrieve.

Folder.GetList Method

Retrieves a list of FolderData objects based upon the supplied criteria.

C#

```
public List<FolderData> GetList(Criteria<FolderProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<FolderProperty>) As List<FolderData>
```

Parameters

Parameters	Description
Criteria<FolderProperty> criteria	Criteria used to retrieve folders.

Folder.Update Method

Updates an existing folder in the CMS.

C#

```
public void Update(FolderData folderData);
```

Visual Basic

```
Public Function Update(folderData As FolderData) As void
```

Parameters

Parameters	Description
FolderData folderData	The FolderData object to update.

Ektron.Cms.Framework.Messaging Namespace

This is namespace Ektron.Cms.Framework.Messaging.

Classes

	Name	Description
	CmsMessage (see page 1052)	Api class for managing Messages in the CMS.
	CmsMessageType (see page 1056)	Api class for managing MessageTypes in the Cms (see page 7).

Classes

The following table lists classes in this documentation.

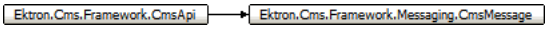
Classes

	Name	Description
	CmsMessage (see page 1052)	Api class for managing Messages in the CMS.
	CmsMessageType (see page 1056)	Api class for managing MessageTypes in the Cms (see page 7).

CmsMessage Class

Api class for managing Messages in the CMS.

Class Hierarchy



C#

```
public class CmsMessage : CmsApi;
```

Visual Basic

```
Public Class CmsMessage
Inherits CmsApi
```

File

CmsMessage.cs

CmsMessage Constructor

This is the overview for the CmsMessage constructor overload.

Overload List

	Name	Description
	CmsMessage.CmsMessage () (see page 1053)	This is CmsMessage, a member of class CmsMessage.
	CmsMessage.CmsMessage (ApiAccessMode) (see page 1053)	

CmsMessage.CmsMessage Constructor ()

C#

```
public CmsMessage();
```

Visual Basic

```
Public Sub New()
```

Description

This is CmsMessage, a member of class CmsMessage.

CmsMessage.CmsMessage Constructor (ApiAccessMode)

C#

```
public CmsMessage(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

CmsMessage Fields

The fields of the CmsMessage class are listed here.

Private Fields

	Name	Description
	_messageService (see page 1053)	This is _messageService, a member of class CmsMessage.

CmsMessage._messageService Field

C#

```
private ICmsMessage _messageService;
```

Visual Basic

```
Private _messageService As ICmsMessage
```

Description

This is _messageService, a member of class CmsMessage.

CmsMessage Methods

The methods of the CmsMessage class are listed here.

Public Methods

	Name	Description
	Add (see page 1054)	Adds a new message based on information in an CmsMessageData object. message.Id will be populated with the newly created ID.
	Delete (see page 1054)	Deletes a message in the CMS.
	GetDefaultItemByType (see page 1054)	Gets the default message for a given message type in the CMS. All available languages of the message are returned as part of the CmsMessageDataCollection.
	GetItem (see page 1055)	Retrieves a single CmsMessageData object from the CMS in the current language.
	GetItemCollection (see page 1055)	Gets a single message from the CMS in all available languages.
	GetList (see page 1055)	Retrieves a list of CmsMessageData objects based upon the supplied criteria.
	Update (see page 1055)	Updates an existing message in the CMS.

CmsMessage.Add Method

Adds a new message based on information in an CmsMessageData object. message.Id will be populated with the newly created ID.

C#

```
public void Add(CmsMessageData message);
```

Visual Basic

```
Public Function Add(message As CmsMessageData) As void
```

Parameters

Parameters	Description
CmsMessageData message	The CmsMessageData object to add.

CmsMessage.Delete Method

Deletes a message in the CMS.

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	The ID of the message to delete

CmsMessage.GetDefaultItemByType Method

Gets the default message for a given message type in the CMS. All available languages of the message are returned as part of the CmsMessageDataCollection.

C#

```
public CmsMessageDataCollection GetDefaultItemByType(long messageTypeId);
```

Visual Basic

```
Public Function GetDefaultItemByType(messageTypeId As long) As CmsMessageDataCollection
```

Parameters

Parameters	Description
long messageTypeId	The ID of the message type to retrieve the default message for.

CmsMessage.GetItem Method

Retrieves a single CmsMessageData object from the CMS in the current language.

C#

```
public CmsMessageData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As CmsMessageData
```

Parameters

Parameters	Description
long id	The ID of the message to retrieve.

CmsMessage.GetItemCollection Method

Gets a single message from the CMS in all available languages.

C#

```
public CmsMessageDataCollection GetItemCollection(long id);
```

Visual Basic

```
Public Function GetItemCollection(id As long) As CmsMessageDataCollection
```

Parameters

Parameters	Description
long id	The ID of the message to retrieve.

CmsMessage.GetList Method

Retrieves a list of CmsMessageData objects based upon the supplied criteria.

C#

```
public List<CmsMessageData> GetList(Criteria<CmsMessageProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<CmsMessageProperty>) As List<CmsMessageData>
```

Parameters

Parameters	Description
Criteria<CmsMessageProperty> criteria	Criteria used to retrieve messages.

CmsMessage.Update Method

Updates an existing message in the CMS.

C#

```
public void Update(CmsMessageData message);
```

Visual Basic

```
Public Function Update(message As CmsMessageData) As void
```

Parameters

Parameters	Description
CmsMessageData message	The CmsMessageData object to update.

CmsMessage Properties

The properties of the CmsMessage class are listed here.

Protected Properties

	Name	Description
	MessageService (see page 1056)	Gets instance to ICmsMessage.

CmsMessage.MessageService Property

Gets instance to ICmsMessage.

C#

```
protected ICmsMessage MessageService;
```

Visual Basic

```
Protected Property MessageService() As ICmsMessage
```

CmsMessageType Class

Api class for managing MessageTypes in the Cms (see page 7).

Class Hierarchy

C#

```
public class CmsMessageType : CmsApi;
```

Visual Basic

```
Public Class CmsMessageType
Inherits CmsApi
```

File

CmsMessageType.cs

CmsMessageType Constructor

This is the overview for the CmsMessageType constructor overload.

Overload List

	Name	Description
	CmsMessageType.CmsMessageType () (see page 1057)	This is CmsMessageType, a member of class CmsMessageType.

	CmsMessageType.CmsMessageType (ApiAccessMode) (see page 1057)	

CmsMessageType.CmsMessageType Constructor ()

C#

```
public CmsMessageType();
```

Visual Basic

```
Public Sub New()
```

Description

This is CmsMessageType, a member of class CmsMessageType.

CmsMessageType.CmsMessageType Constructor (ApiAccessMode)

C#

```
public CmsMessageType(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

CmsMessageType Fields

The fields of the CmsMessageType class are listed here.

Private Fields

	Name	Description
	_messageTypeService (see page 1057)	This is _messageTypeService, a member of class CmsMessageType.

CmsMessageType._messageTypeService Field

C#

```
private ICmsMessageType _messageTypeService;
```

Visual Basic

```
Private _messageTypeService As ICmsMessageType
```

Description

This is _messageTypeService, a member of class CmsMessageType.

CmsMessageType Methods

The methods of the CmsMessageType class are listed here.

Public Methods

	Name	Description
	Add (see page 1058)	Adds a new message type based on information in an CmsMessageTypeData object. messageType.Id will be populated with the newly created ID.

	Delete (see page 1058)	Deletes a message type in the CMS.
	GetItem (see page 1058)	Retrieves a single CmsMessageTypeData object from the CMS.
	GetList (see page 1058)	Retrieves a list of CmsMessageTypeData objects based upon the supplied criteria.
	GetTokenList (see page 1059)	Returns the list of applicable replacement tokens for a given MessageTypeId.
	Update (see page 1059)	Updates an existing message type in the CMS.

CmsMessageType.Add Method

Adds a new message type based on information in an CmsMessageTypeData object. messageType.Id will be populated with the newly created ID.

C#

```
public void Add(CmsMessageTypeData messageType);
```

Visual Basic

```
Public Function Add(messageType As CmsMessageTypeData) As void
```

Parameters

Parameters	Description
message	The CmsMessageData object to add.

CmsMessageType.Delete Method

Deletes a message type in the CMS.

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	The ID of the message to delete

CmsMessageType.GetItem Method

Retrieves a single CmsMessageTypeData object from the CMS.

C#

```
public CmsMessageTypeData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As CmsMessageTypeData
```

Parameters

Parameters	Description
long id	The ID of the message type to retrieve.

CmsMessageType.GetList Method

Retrieves a list of CmsMessageTypeData objects based upon the supplied criteria.

C#

```
public List<CmsMessageTypeData> GetList(Criteria<CmsMessageTypeProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<CmsMessageTypeProperty>) As
List<CmsMessageTypeData>
```

Parameters

Parameters	Description
Criteria<CmsMessageTypeProperty> criteria	Criteria used to retrieve message types.

CmsMessageType.GetTokenList Method

Returns the list of applicable replacement tokens for a given MessageTypeId.

C#

```
public List<string> GetTokenList(long messageTypeId);
```

Visual Basic

```
Public Function GetTokenList(messageTypeId As long) As List<string>
```

Parameters

Parameters	Description
long messageTypeId	ID of Message Type to retrieve tokens for.

CmsMessageType.Update Method

Updates an existing message type in the CMS.

C#

```
public void Update(CmsMessageTypeData messageType);
```

Visual Basic

```
Public Function Update(messageType As CmsMessageTypeData) As void
```

Parameters

Parameters	Description
message	The CmsMessageTypeData object to update.

CmsMessageType Properties

The properties of the CmsMessageType class are listed here.

Protected Properties

	Name	Description
	MessageTypeService (see page 1059)	Gets instance to ICmsMessageType.

CmsMessageType.MessageTypeService Property

Gets instance to ICmsMessageType.

C#

```
protected ICmsMessageType MessageTypeService;
```

Visual Basic

```
Protected Property MessageTypeService() As ICmsMessageType
```

Ektron.Cms.Framework.Notifications Namespace

This is namespace Ektron.Cms.Framework.Notifications.

Classes

	Name	Description
	Notification (see page 1060)	Api class for sending out notifications in the CMS.
	NotificationAgentSetting (see page 1062)	Api class for managing settings for Notification (see page 1060) Agent settings.
	NotificationPreference (see page 1064)	Api class for managing user notification preferences in the CMS.
	UserNotificationSetting (see page 1067)	Api class for managing user notification settings in the CMS. UserNotificationSettigns are user settings per notification agent. For example, to recieve notifications via Email, the user would need to provide hisher email address, etc.

Classes

The following table lists classes in this documentation.

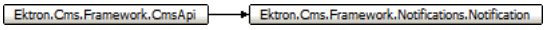
Classes

	Name	Description
	Notification (see page 1060)	Api class for sending out notifications in the CMS.
	NotificationAgentSetting (see page 1062)	Api class for managing settings for Notification (see page 1060) Agent settings.
	NotificationPreference (see page 1064)	Api class for managing user notification preferences in the CMS.
	UserNotificationSetting (see page 1067)	Api class for managing user notification settings in the CMS. UserNotificationSettigns are user settings per notification agent. For example, to recieve notifications via Email, the user would need to provide hisher email address, etc.

Notification Class

Api class for sending out notifications in the CMS.

Class Hierarchy



C#

```
public class Notification : CmsApi;
```

Visual Basic

```
Public Class Notification
Inherits CmsApi
```

File

NotificationApi.cs

Notification.Notification Constructor

C#

```
public Notification(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

Notification Methods

The methods of the Notification class are listed here.

Public Methods

	Name	Description
	Send (see page 1061)	This is the overview for the Send method overload.

Send Method

This is the overview for the Send method overload.

Overload List

	Name	Description
	Notification.Send (ActivityData) (see page 1061)	Sends notifications based upon the supplied activity.
	Notification.Send (NotificationMessageData) (see page 1061)	Sends notifications based upon supplie dnotification message data.

Notification.Send Method (ActivityData)

Sends notifications based upon the supplied activity.

C#

```
public void Send(ActivityData activity);
```

Visual Basic

```
Public Function Send(activity As ActivityData) As void
```

Parameters

Parameters	Description
ActivityData activity	The activity that has occured and will result in notifications being sent. The Activity.Message property is ignored and will be retrieved based upon CMS message settings.

Notification.Send Method (NotificationMessageData)

Sends notifications based upon supplie dnotification message data.

C#

```
public void Send(NotificationMessageData notificationMessage);
```

Visual Basic

```
Public Function Send(notificationMessage As NotificationMessageData) As void
```

Parameters

Parameters	Description
NotificationMessageData notificationMessage	data object defining message to send and the user to send it to.

NotificationAgentSetting Class

Api class for managing settings for Notification (see page 1060) Agent settings.

Class Hierarchy

C#

```
public class NotificationAgentSetting : CmsApi;
```

Visual Basic

```
Public Class NotificationAgentSetting
Inherits CmsApi
```

File

NotificationAgentSetting.cs

NotificationAgentSetting.NotificationAgentSetting Constructor

C#

```
public NotificationAgentSetting(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

NotificationAgentSetting Methods

The methods of the NotificationAgentSetting class are listed here.

Public Methods

	Name	Description
	Add (see page 1062)	Adds a new notification agent to the system.
	Delete (see page 1063)	Deletes a notification agent settings from the CMS.
	GetAgent (see page 1063)	Returns actuual NotificationAgent based upon Agent Id.
	GetItem (see page 1063)	Retrieves a notifcation agent setting object.
	GetList (see page 1063)	Retrieves a list of Notification (see page 1060) Agent settings in the CMS based upon the supplied criteria.
	GetRegisteredAgentList (see page 1064)	Retrieves a list of the Notification (see page 1060) Agents registered in the web.config.
	Update (see page 1064)	Updates an existing notification agent settings.

NotificationAgentSetting.Add Method

Adds a new notification agent to the system.

C#

```
public void Add(NotificationAgentData agentData);
```

Visual Basic

```
Public Function Add(agentData As NotificationAgentData) As void
```

Parameters

Parameters	Description
NotificationAgentData agentData	Agent settings to save.

NotificationAgentSetting.Delete Method

Deletes a notification agent settings from the CMS.

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	Id of agent to delete.

NotificationAgentSetting.GetAgent Method

Returns actuual NotificationAgent based upon Agent Id.

C#

```
public NotificationAgent GetAgent(long id);
```

Visual Basic

```
Public Function GetAgent(id As long) As NotificationAgent
```

Parameters

Parameters	Description
long id	Id of agent to retrieve.

NotificationAgentSetting.GetItem Method

Retrieves a notifcation agent setting object.

C#

```
public NotificationAgentData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As NotificationAgentData
```

Parameters

Parameters	Description
long id	Id of agent ot retrieve settings for.

NotificationAgentSetting.GetList Method

Retrieves a list of Notification (see page 1060) Agent settings in the CMS based upon the supplied criteria.

C#

```
public List<NotificationAgentData> GetList(Criteria<NotificationAgentProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<NotificationAgentProperty>) As
List<NotificationAgentData>
```

Parameters

Parameters	Description
Criteria<NotificationAgentProperty> criteria	criteira by which to retrieve agent settings.

NotificationAgentSetting.GetRegisteredAgentList Method

Retrieves a list of the Notification (see page 1060) Agents registered in the web.config.

C#

```
public List<NotificationAgent> GetRegisteredAgentList();
```

Visual Basic

```
Public Function GetRegisteredAgentList() As List<NotificationAgent>
```

Parameters

Parameters	Description
criteria	

NotificationAgentSetting.Update Method

Updates an existing notification agent settings.

C#

```
public void Update(NotificationAgentData agentData);
```

Visual Basic

```
Public Function Update(agentData As NotificationAgentData) As void
```

Parameters

Parameters	Description
NotificationAgentData agentData	Agent settings to save.

NotificationPreference Class

Api class for managing user notification preferences in the CMS.

Class Hierarchy

C#

```
public class NotificationPreference : CmsApi;
```

Visual Basic

```
Public Class NotificationPreference
Inherits CmsApi
```

File

NotificationPreferenceApi.cs

NotificationPreference.NotificationPreference Constructor

C#

```
public NotificationPreference(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

NotificationPreference Methods

The methods of the NotificationPreference class are listed here.

Public Methods

	Name	Description
	Add (see page 1065)	Saves a new notification preference.
	Delete (see page 1065)	Deletes a notification preference.
	GetDefaultPreferenceList (see page 1065)	Returns a list of default notification preferences based upon the supplied criteira.
	GetItem (see page 1066)	Returns a single notification preference by Id.
	GetList (see page 1066)	Returns a list of user notification preferences based upon the supplied criteira.
	GetUserNotificationListForActivity (see page 1066)	Returns a list of users and their nortfication settings for a given activity ocurrence.
	SaveUserPreferences (see page 1066)	Saves a set of notification preferences for a given user. Preferences marked as added will be added and those marked as deleted will be removed.
	Update (see page 1067)	Updates an existing notification preference.

NotificationPreference.Add Method

Saves a new notification preference.

C#

```
public void Add(NotificationPreferenceData preferenceData);
```

Visual Basic

```
Public Function Add(preferenceData As NotificationPreferenceData) As void
```

Parameters

Parameters	Description
NotificationPreferenceData preferenceData	preference data to save.

NotificationPreference.Delete Method

Deletes a notification preference.

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	Id of NotificationPreference (see page 1064) to delete.

NotificationPreference.GetDefaultPreferenceList Method

Returns a list of default notification preferences based upon the supplied criteira.

C#

```
public List<NotificationPreferenceData>
GetDefaultPreferenceList(Criteria<NotificationPreferenceProperty> criteria);
```

Visual Basic

```
Public Function GetDefaultPreferenceList(criteria As Criteria<NotificationPreferenceProperty>)
As List<NotificationPreferenceData>
```

Parameters

Parameters	Description
Criteria<NotificationPreferenceProperty> criteria	criteria used to determine which default preferences to return.

NotificationPreference.GetItem Method

Returns a single notification preference by Id.

C#

```
public NotificationPreferenceData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As NotificationPreferenceData
```

Parameters

Parameters	Description
long id	Id of NotificationPreference (see page 1064) to return.

NotificationPreference.GetList Method

Returns a list of user notification preferences based upon the supplied criteira.

C#

```
public List<NotificationPreferenceData> GetList(Criteria<NotificationPreferenceProperty>
criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<NotificationPreferenceProperty>) As
List<NotificationPreferenceData>
```

Parameters

Parameters	Description
Criteria<NotificationPreferenceProperty> criteria	criteria used to determine which preferences to return.

NotificationPreference.GetUserNotificationListForActivity Method

Returns a list of users and their nortfication settings for a given activity ocurrence.

C#

```
public List<UserNotificationData> GetUserNotificationListForActivity(ActivityData activity,
PagingInfo paging);
```

Visual Basic

```
Public Function GetUserNotificationListForActivity(activity As ActivityData, paging As
PagingInfo) As List<UserNotificationData>
```

Parameters

Parameters	Description
ActivityData activity	Activity (see page 996) defintion to retrieve notifications for.
PagingInfo paging	paging information for the retrieval.

NotificationPreference.SaveUserPreferences Method

Saves a set of notification preferences for a given user. Preferences marked as added will be added and those marked as

deleted will be removed.

C#

```
public void SaveUserPreferences(List<NotificationPreferenceData> preferenceList);
```

Visual Basic

```
Public Function SaveUserPreferences(preferenceList As List<NotificationPreferenceData>) As void
```

Parameters

Parameters	Description
preferenceData	preference data to save.

NotificationPreference.Update Method

Updates an existing notification preference.

C#

```
public void Update(NotificationPreferenceData preferenceData);
```

Visual Basic

```
Public Function Update(preferenceData As NotificationPreferenceData) As void
```

Parameters

Parameters	Description
NotificationPreferenceData preferenceData	preference data to save.

UserNotificationSetting Class

Api class for managing user notification settings in the CMS. UserNotificationSettigns are user settings per notification agent. For example, to recieve notifications via Email, the user would need to provide hisher email address, etc.

Class Hierarchy

C#

```
public class UserNotificationSetting : CmsApi;
```

Visual Basic

```
Public Class UserNotificationSetting
Inherits CmsApi
```

File

UserNotificationSettingApi.cs

UserNotificationSetting.UserNotificationSetting Constructor

C#

```
public UserNotificationSetting(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

UserNotificationSetting Methods

The methods of the UserNotificationSetting class are listed here.

Public Methods

	Name	Description
	Add (see page 1068)	Adds a new user notification setting.
	Delete (see page 1068)	Deletes a User (see page 950) Notification (see page 1060) Setting.
	GetItem (see page 1068)	Returns a single UserNotificationSettingData by Id.
	GetList (see page 1069)	Returns a list of UserNotificationSettingData based upon the supplied criteira.
	Update (see page 1069)	Updates an existing user notification setting.
	VerifyValidationCode (see page 1069)	Verifies that the validation code is correct and enables users notfication settings.

UserNotificationSetting.Add Method

Adds a new user notification setting.

C#

```
public void Add(UserNotificationSettingData userSettingData);
```

Visual Basic

```
Public Function Add(userSettingData As UserNotificationSettingData) As void
```

Parameters

Parameters	Description
UserNotificationSettingData userSettingData	user setting data to save.

UserNotificationSetting.Delete Method

Deletes a User (see page 950) Notification (see page 1060) Setting.

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	Id of User (see page 950) Notification (see page 1060) Setting to delete.

UserNotificationSetting.GetItem Method

Returns a single UserNotificationSettingData by Id.

C#

```
public UserNotificationSettingData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As UserNotificationSettingData
```

Parameters

Parameters	Description
long id	Id of UserNotificationSettingData to return.

UserNotificationSetting.GetList Method

Returns a list of UserNotificationSettingData based upon the supplied criteira.

C#

```
public List<UserNotificationSettingData> GetList(Criteria<UserNotificationSettingProperty>
criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<UserNotificationSettingProperty>) As
List<UserNotificationSettingData>
```

Parameters

Parameters	Description
Criteria<UserNotificationSettingProperty> criteria	criteria used to determine which UserNotificationSettingData to return.

UserNotificationSetting.Update Method

Updates an existing user notification setting.

C#

```
public void Update(UserNotificationSettingData userSettingData);
```

Visual Basic

```
Public Function Update(userSettingData As UserNotificationSettingData) As void
```

Parameters

Parameters	Description
UserNotificationSettingData userSettingData	user setting data to save.

UserNotificationSetting.VerifyValidationCode Method

Verifies that the validation code is correct and enables users notfication settings.

C#

```
public bool VerifyValidationCode(long userSettingId, string verificationCode);
```

Visual Basic

```
Public Function VerifyValidationCode(userSettingId As long, verificationCode As string) As bool
```

Parameters

Parameters	Description
long userSettingId	Id of user notification setting.
string verificationCode	Verification code to validate.

Returns

true if the verification code is correct.

Ektron.Cms.Framework.SocialNetworking Namespace

This is namespace Ektron.Cms.Framework.SocialNetworking.

Classes

	Name	Description
	MicroMessage (see page 1070)	This is class Ektron.Cms.Framework.SocialNetworking.MicroMessage.

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	MicroMessage (see page 1070)	This is class Ektron.Cms.Framework.SocialNetworking.MicroMessage.

MicroMessage Class

Class Hierarchy

C#

```
public class MicroMessage : CmsApi;
```

Visual Basic

```
Public Class MicroMessage
Inherits CmsApi
```

File

MicroMessage.cs

Description

This is class Ektron.Cms.Framework.SocialNetworking.MicroMessage.

MicroMessage Constructor

This is the overview for the MicroMessage constructor overload.

Overload List

	Name	Description
	MicroMessage.MicroMessage () (see page 1070)	This is MicroMessage, a member of class MicroMessage.
	MicroMessage.MicroMessage (ApiAccessMode) (see page 1071)	This is MicroMessage, a member of class MicroMessage.

MicroMessage.MicroMessage Constructor ()

C#

```
public MicroMessage();
```

Visual Basic

```
Public Sub New()
```

Description

This is MicroMessage, a member of class MicroMessage.

MicroMessage.MicroMessage Constructor (ApiAccessMode)

C#

```
public MicroMessage(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Description

This is MicroMessage, a member of class MicroMessage.

MicroMessage Fields

The fields of the MicroMessage class are listed here.

Private Fields

	Name	Description
	_MicroMessageBL (see page 1071)	This is _MicroMessageBL, a member of class MicroMessage.

MicroMessage._MicroMessageBL Field

C#

```
private IMicroMessage _MicroMessageBL;
```

Visual Basic

```
Private _MicroMessageBL As IMicroMessage
```

Description

This is _MicroMessageBL, a member of class MicroMessage.

MicroMessage Methods

The methods of the MicroMessage class are listed here.

Public Methods

	Name	Description
	Add (see page 1072)	Adds a new message.
	AddReply (see page 1072)	Adds a new reply.
	Delete (see page 1072)	Deletes a record from the database.
	GetColleagueMessageList (see page 1072)	This is the overview for the GetColleagueMessageList method overload.
	GetItem (see page 1073)	Gets a single record from the database.
	GetList (see page 1073)	Gets data from the database using the criteria object.
	GetReplyList (see page 1074)	Gets a list of replies for a Message.
	GetSearchList (see page 1074)	Gets the search results for test search.
	GetTimeLineMessageList (see page 1074)	This is the overview for the GetTimeLineMessageList method overload.
	GetUserMessageList (see page 1075)	Gets a list of messages that the user has sent.

	IsFullTextSearchInstalled (see page 1075)	Checks if full text search is installed on the database.
	IsSpam (see page 1075)	This is IsSpam, a member of class MicroMessage.

MicroMessage.Add Method

Adds a new message.

C#

```
public void Add(MicroMessageData data);
```

Visual Basic

```
Public Function Add(data As MicroMessageData) As void
```

Parameters

Parameters	Description
MicroMessageData data	The micro message data.

MicroMessage.AddReply Method

Adds a new reply.

C#

```
public void AddReply(long parentMessageId, string messageText);
```

Visual Basic

```
Public Function AddReply(parentMessageId As long, messageText As string) As void
```

Parameters

Parameters	Description
long parentMessageId	The id of the message to add a reply to.
string messageText	The text of the reply

MicroMessage.Delete Method

Deletes a record from the database.

C#

```
public void Delete(long Id);
```

Visual Basic

```
Public Function Delete(Id As long) As void
```

Parameters

Parameters	Description
long Id	The id of the message to be deleted.

GetColleagueMessageList Method

This is the overview for the GetColleagueMessageList method overload.

Overload List

	Name	Description
	MicroMessage.GetColleagueMessageList (PagingInfo) (see page 1073)	Gets a list of only colleagues Messages.
	MicroMessage.GetColleagueMessageList (PagingInfo, long) (see page 1073)	Gets a list of only colleagues Messages.

MicroMessage.GetColleagueMessageList Method (PagingInfo)

Gets a list of only colleagues Messages.

C#

```
public List<MicroMessageData> GetColleagueMessageList(PagingInfo paging);
```

Visual Basic

```
Public Function GetColleagueMessageList(paging As PagingInfo) As List<MicroMessageData>
```

Parameters

Parameters	Description
PagingInfo paging	

MicroMessage.GetColleagueMessageList Method (PagingInfo, long)

Gets a list of only colleagues Messages.

C#

```
public List<MicroMessageData> GetColleagueMessageList(PagingInfo paging, long callerUserId);
```

Visual Basic

```
Public Function GetColleagueMessageList(paging As PagingInfo, callerUserId As long) As
List<MicroMessageData>
```

Parameters

Parameters	Description
PagingInfo paging	
long callerUserId	

MicroMessage.GetItem Method

Gets a single record from the database.

C#

```
public MicroMessageData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As MicroMessageData
```

Parameters

Parameters	Description
long id	The id of the message.

MicroMessage.GetList Method

Gets data from the database using the criteria object.

C#

```
public List<MicroMessageData> GetList(Criteria<MicroMessageProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<MicroMessageProperty>) As List<MicroMessageData>
```

Parameters

Parameters	Description
Criteria<MicroMessageProperty> criteria	

MicroMessage.GetReplyList Method

Gets a list of replies for a Message.

C#

```
public List<MicroMessageData> GetReplyList(long parentMessageId);
```

Visual Basic

```
Public Function GetReplyList(parentMessageId As long) As List<MicroMessageData>
```

Parameters

Parameters	Description
long parentMessageId	The id of the message that has replies.

Returns

Returns a list of replies for the message.

MicroMessage.GetSearchList Method

Gets the search results for test search.

C#

```
public List<MicroMessageData> GetSearchList(PagingInfo paging, String searchText);
```

Visual Basic

```
Public Function GetSearchList(paging As PagingInfo, searchText As String) As
List<MicroMessageData>
```

Parameters

Parameters	Description
PagingInfo paging	
String searchText	

GetTimeLineMessageList Method

This is the overview for the GetTimeLineMessageList method overload.

Overload List

	Name	Description
	MicroMessage.GetTimeLineMessageList (PagingInfo) (see page 1074)	Gets a filtered list of messages that the requesting user can see. The list includes public and colleagues messages.
	MicroMessage.GetTimeLineMessageList (PagingInfo, long) (see page 1075)	Gets a filtered list of messages that the requesting user can see. The list includes public and colleagues messages.

MicroMessage.GetTimeLineMessageList Method (PagingInfo)

Gets a filtered list of messages that the requesting user can see. The list includes public and colleagues messages.

C#

```
public List<MicroMessageData> GetTimeLineMessageList(PagingInfo paging);
```

Visual Basic

```
Public Function GetTimeLineMessageList(paging As PagingInfo) As List<MicroMessageData>
```

Parameters

Parameters	Description
PagingInfo paging	

Returns

Returns a list of for the user messages.

MicroMessage.GetTimeLineMessageList Method (PagingInfo, long)

Gets a filtered list of messages that the requesting user can see. The list includes public and colleagues messages.

C#

```
public List<MicroMessageData> GetTimeLineMessageList(PagingInfo paging, long callerUserId);
```

Visual Basic

```
Public Function GetTimeLineMessageList(paging As PagingInfo, callerUserId As long) As
List<MicroMessageData>
```

Parameters

Parameters	Description
PagingInfo paging	
long callerUserId	

MicroMessage.GetUserMessageList Method

Gets a list of messages that the user has sent.

C#

```
public List<MicroMessageData> GetUserMessageList(PagingInfo paging, long userId);
```

Visual Basic

```
Public Function GetUserMessageList(paging As PagingInfo, userId As long) As
List<MicroMessageData>
```

Parameters

Parameters	Description
PagingInfo paging	
callerUserId	

Returns

Returns a list of for the user messages.

MicroMessage.IsFullTextSearchInstalled Method

Checks if full text search is installed on the database.

C#

```
public bool IsFullTextSearchInstalled();
```

Visual Basic

```
Public Function IsFullTextSearchInstalled() As bool
```

MicroMessage.IsSpam Method

C#

```
public Boolean IsSpam(EkEnumeration.MicroMessageSpamControlType spamControlType, int
SpamTimeSpan, MicroMessageData data);
```

Visual Basic

```
Public Function IsSpam(spamControlType As EkEnumeration.MicroMessageSpamControlType,
SpamTimeSpan As Integer, data As MicroMessageData) As Boolean
```

Description

This is IsSpam, a member of class MicroMessage.

MicroMessage Properties

The properties of the MicroMessage class are listed here.

Private Properties

	Name	Description
	GetInstance (see page 1076)	This is GetInstance, a member of class MicroMessage.

MicroMessage.GetInstance Property

C#

```
private IMicroMessage GetInstance;
```

Visual Basic

```
Private Property GetInstance() As IMicroMessage
```

Description

This is GetInstance, a member of class MicroMessage.

Ektron.Cms.Framework.Users Namespace

This is namespace Ektron.Cms.Framework.Users.

Classes

	Name	Description
	User (see page 1076)	This is class Ektron.Cms.Framework.Users.User.

Classes

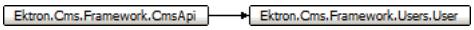
The following table lists classes in this documentation.

Classes

	Name	Description
	User (see page 1076)	This is class Ektron.Cms.Framework.Users.User.

User Class

Class Hierarchy



C#

```
public class User : CmsApi;
```

Visual Basic

```
Public Class User
Inherits CmsApi
```

File

User.cs

Description

This is class Ektron.Cms.Framework.Users.User.

User Constructor

This is the overview for the User constructor overload.

Overload List

	Name	Description
	User.User () (see page 1077)	constructor
	User.User (ApiAccessMode) (see page 1077)	constructor

User.User Constructor ()

constructor

C#

```
public User();
```

Visual Basic

```
Public Sub New()
```

User.User Constructor (ApiAccessMode)

constructor

C#

```
public User(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

User Fields

The fields of the User class are listed here.

Private Fields

	Name	Description
	_userService (see page 1078)	This is _userService, a member of class User.

User._userService Field

C#

```
private IUser _userService;
```

Visual Basic

```
Private _userService As IUser
```

Description

This is _userService, a member of class User.

User Methods

The methods of the User class are listed here.

Public Methods

	Name	Description
	Add (see page 1078)	Adds a new user, with details from the supplied UserData object.
	Delete (see page 1078)	Deletes an existing user.
	GetItem (see page 1078)	Returns details of a user.
	GetList (see page 1079)	Returns a list of users that fit the supplied criteria.
	Update (see page 1079)	Updates a user, with details from the supplied UserData object.

User.Add Method

Adds a new user, with details from the supplied UserData object.

C#

```
public void Add(UserData userData);
```

Visual Basic

```
Public Function Add(userData As UserData) As void
```

Parameters

Parameters	Description
UserData userData	

User.Delete Method

Deletes an existing user.

C#

```
public void Delete(long id);
```

Visual Basic

```
Public Function Delete(id As long) As void
```

Parameters

Parameters	Description
long id	the identifier of the user to delete

User.GetItem Method

Returns details of a user.

C#

```
public UserData GetItem(long id);
```

Visual Basic

```
Public Function GetItem(id As long) As UserData
```

Parameters

Parameters	Description
long id	the identifier of the user to get

Returns

user details in a UserData object

User.GetList Method

Returns a list of users that fit the supplied criteria.

C#

```
public List<UserData> GetList(Criteria<UserProperty> criteria);
```

Visual Basic

```
Public Function GetList(criteria As Criteria<UserProperty>) As List<UserData>
```

Parameters

Parameters	Description
Criteria<UserProperty> criteria	used to specify, or filter, the users to return

Returns

a list of users

User.Update Method

Updates a user, with details from the supplied UserData object.

C#

```
public void Update(UserData userData);
```

Visual Basic

```
Public Function Update(userData As UserData) As void
```

Parameters

Parameters	Description
UserData userData	

User Properties

The properties of the User class are listed here.

Protected Properties

	Name	Description
	UserService (see page 1079)	returns an instance of the business objects user service.

User.UserService Property

returns an instance of the business objects user service.

C#

```
protected IUser UserService;
```

Visual Basic

```
Protected Property UserService() As IUser
```

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	CmsApi (see page 1080)	Base API (see page 669) class for all Ektron CMS APIs.

CmsApi Class

Base API (see page 669) class for all Ektron CMS APIs.

Class Hierarchy

Ektron.Cms.Framework.CmsApi

C#

```
public abstract class CmsApi;
```

Visual Basic

```
Public abstract Class CmsApi
```

File

CmsApi.cs

CmsApi Constructor

This is the overview for the CmsApi constructor overload.

Overload List

	Name	Description
	CmsApi.CmsApi () (see page 1080)	Default API (see page 669) constructor. By defualt the API (see page 669), runs as the currently logged in user.
	CmsApi.CmsApi (ApiAccessMode) (see page 1081)	

CmsApi.CmsApi Constructor ()

Default API (see page 669) constructor. By defualt the API (see page 669), runs as the currently logged in user.

C#

```
public CmsApi();
```

Visual Basic

```
Public Sub New()
```

CmsApi.CmsApi Constructor (ApiAccessMode)

C#

```
public CmsApi(ApiAccessMode mode);
```

Visual Basic

```
Public Sub New(mode As ApiAccessMode)
```

Parameters

Parameters	Description
ApiAccessMode mode	Sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

CmsApi Properties

The properties of the CmsApi class are listed here.

Public Properties

	Name	Description
	ApiMode (see page 1081)	Gets or sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.
	ContentLanguage (see page 1081)	Gets or sets the current content language.
	InPreviewMode (see page 1081)	Gets or sets the current Preview Mode. Returns true if the site is in site preview mode.
	RequestInformation (see page 1082)	Gets the current Request Information.
	SitePath (see page 1082)	Gets the site path.
	UserId (see page 1082)	Gets or sets the currently logged in user ID.

CmsApi.ApiMode Property

Gets or sets the current API (see page 669) access mode. If set to Admin, the API (see page 669) runs with the permissions of an administrator.

C#

```
public ApiAccessMode ApiMode;
```

Visual Basic

```
Public Property ApiMode() As ApiAccessMode
```

CmsApi.ContentLanguage Property

Gets or sets the current content language.

C#

```
public int ContentLanguage;
```

Visual Basic

```
Public Property ContentLanguage() As Integer
```

CmsApi.InPreviewMode Property

Gets or sets the current Preview Mode. Returns true if the site is in site preview mode.

C#

```
public bool InPreviewMode;
```

Visual Basic

```
Public Property InPreviewMode() As bool
```

CmsApi.RequestInformation Property

Gets the current Request Information.

C#

```
public EkRequestInformation RequestInformation;
```

Visual Basic

```
Public Property RequestInformation() As EkRequestInformation
```

CmsApi.SitePath Property

Gets the site path.

C#

```
public string SitePath;
```

Visual Basic

```
Public Property SitePath() As string
```

Example

An example of using this property can be found in the Workarea source file `CMS400Demo\Workarea\isearch.aspx.vb`.

CmsApi.UserId Property

Gets or sets the currently logged in user ID.

C#

```
public long UserId;
```

Visual Basic

```
Public Property UserId() As long
```

Returns

Long

Structs, Records, Enums

The following table lists structs, records, enums in this documentation.

Enumerations

	Name	Description
	ApiAccessMode (see page 1082)	Enumeration used to indicate the access mode when using an API (see page 669) class.

Ektron.Cms.Framework.ApiAccessMode Enumeration

Enumeration used to indicate the access mode when using an API (see page 669) class.

C#

```
public enum ApiAccessMode {
  LoggedInUser = 0,
```

```
  Admin = 1
}
```

Visual Basic

```
Public Enum ApiAccessMode
  LoggedInUser = 0
  Admin = 1
End Enum
```

File

ApiAccessMode.cs

Members

Members	Description
LoggedInUser = 0	Indicates that the API (see page 669) should run as the logged in user. This is the default setting.
Admin = 1	Indicates that the API (see page 669) should run as an administrator.

Web Services API

The Web Services API exposes a method's functionality for use with SOAP over HTTP. For additional information on the Web Services, see the "Web Services" section of the Developer Manual.

Namespaces

Name	Description
Services (see page 1083)	This is namespace Ektron.Services.

Ektron.Services Namespace

This is namespace Ektron.Services.

Namespaces

Name	Description
Task (see page 1085)	This is namespace Ektron.Services.Task.

Classes

	Name	Description
	ActiveDirectoryUser (see page 1103)	This section describes the Web Service methods used with Ektron's Active Directory (see page 196). The Active Directory (see page 196) Web service is a wrapper for the Active Directory (see page 196) API (see page 669): `Ektron.Cms.API.User.ADUser (see page 951).` These methods can be accessed through the ActiveDirectoryUser.asmx file located in:
	Asset (see page 1112)	This section describes the Web Service methods used with Assets. The Assets Web service is a wrapper for the Asset API (see page 669): `Ektron.Cms.API.Content.Asset (see page 822)`. These methods can be accessed through the Asset.asmx file located in:
	Blog (see page 1116)	This section describes the Web Service methods used with blogs. The Blog Web service is a wrapper for the blog API (see page 669): `Ektron.Cms.API.Content.Blogs.` These methods can be accessed through the blog.asmx file located in:
	BusinessRules (see page 1130)	This section describes the Web Service methods used with Ektron's Business Rules. The Business Rules Web service is a wrapper for the Business Rules API (see page 669): `Ektron.Cms.API.BusinessRule (see page 671)`. These methods can be accessed through the BusinessRules.asmx file located in:
	Calendar (see page 1133)	Summary description for Calendar
	Analytics (see page 1133)	This section describes the Web Service methods used with analytics. The Analytics Web service is a wrapper for the Analytics API (see page 669): `Ektron.Cms.API.Analytics.Analytics (see page 745)`. These methods can be accessed through the Analytics.asmx file located in:

	CommunityGroup (see page 1181)	This section describes the Web Service methods used with Ektron's Community (see page 750) Groups. The Community (see page 750) Groups Web service is a wrapper for the Community (see page 750) Groups API (see page 669): `Ektron.Cms.API.Community.CommunityGroup (see page 750)`. These methods can be accessed through the CommunityGroup.asmx file located in:
	Content (see page 1198)	This section describes the Web Service methods used with content. The Content Web service is a wrapper for the Content API (see page 669): `Ektron.Cms.API.Content.Content (see page 840)`. These methods can be accessed through the Content.asmx file located in:
	ContentRating (see page 1214)	This section describes the Web Service methods used with Ektron's Content (see page 1198) Ratings. The Content (see page 1198) Rating Web service is a wrapper for the Content (see page 1198) Rating API (see page 669): `Ektron.Cms.API.Content.ContentRating (see page 866)`. These methods can be accessed through the ContentRating.asmx file located in:
	CustomFields (see page 1216)	This section describes the Web Service methods used with Ektron's Custom Fields. The Custom Fields Web service is a wrapper for the Custom Fields API (see page 669): `Ektron.Cms.API.CustomFields (see page 673)`. These methods can be accessed through the CustomFields.asmx file located in:
	Favorites (see page 1225)	This section describes the Web Service methods used with Ektron's Favorites feature. This feature is a part Ektron's Community (see page 750) Platform. The Favorites Web service is a wrapper for the Favorites API (see page 669): `Ektron.Cms.API.Community.Favorites (see page 763)`. These methods can be accessed through the Favorites.asmx file located in:
	Flag (see page 1231)	This section describes the Web Service methods used with Ektron's Flagging (see page 771) feature. This feature is a part Ektron's Community (see page 750) Platform. The Flagging (see page 771) Web service is a wrapper for the Flagging (see page 771) API (see page 669): `Ektron.Cms.API.Community.Flagging (see page 771)`. These methods can be accessed through the Flag.asmx file located in:
	Folder (see page 1236)	This section describes the Web Service methods used with folder. The Folder Web service is a wrapper for the Folder API (see page 669): `Ektron.Cms.API.Folder (see page 683)`. These methods can be accessed through the Folder.asmx file located in:
	Font (see page 1244)	This section describes the Web Service methods used with Fonts. The Font Web service is a wrapper for the Font API (see page 669): `Ektron.Cms.API.Font`. These methods can be accessed through the Font.asmx file located in:
	Form (see page 1247)	This section describes the Web Service methods used with forms. The Form Web service is a wrapper for the Form API (see page 669): `Ektron.Cms.API.Content.Form (see page 871)`. These methods can be accessed through the form.asmx file located in:
	Friends (see page 1255)	This section describes the Web Service methods used with Ektron's Friends feature. This feature is a part Ektron's Community (see page 750) Platform. The Friends Web service is a wrapper for the Friends API (see page 669): `Ektron.Cms.API.Community.Friends (see page 777)`. These methods can be accessed through the Friends.asmx file located in:
	Library (see page 1269)	This section describes the Web Service methods used with the Library. The Library Web service is a wrapper for the Library API (see page 669): `Ektron.Cms.API.Library (see page 695)`. These methods can be accessed through the Library.asmx file located in:
	MessageBoard (see page 1279)	This section describes the Web Service methods used with Ektron's Message Board feature. This feature is a part Ektron's Community (see page 750) Platform. The MessageBoard Web service is a wrapper for the MessageBoard API (see page 669): `Ektron.Cms.API.Community.MessageBoard (see page 787)`. These methods can be accessed through the MessageBoard.asmx file located in:
	Metadata (see page 1300)	This section describes the Web Service methods used with Metadata. The Metadata Web service is a wrapper for the Metadata API (see page 669): `Ektron.Cms.API.Metadata (see page 714)`. These methods can be accessed through the Metadata.asmx file located in:
	Permissions (see page 1304)	This section describes the Web Service methods used with Permissions. The Permissions Web service is a wrapper for the Permissions API (see page 669): `Ektron.Cms.API.Permissions (see page 721)`. These methods can be accessed through the Permissions.asmx file located in:
	SearchManager (see page 1311)	This section describes the Web Service methods used with the SearchManager. The SearchManager Web service is a wrapper for the SearchManager API (see page 669): `Ektron.Cms.API.Search.SearchManager (see page 899)`. These methods can be accessed through the SearchManager.asmx file located in:
	Site (see page 1316)	This section describes the Web Service methods used with Site functionality. The Site Web service is a wrapper for the Site API (see page 669): `Ektron.Cms.API.Site (see page 725)`. These methods can be accessed through the Site.asmx file located in:
	SiteMap (see page 1324)	This section describes the Web Service methods used with SiteMap. The SiteMap Web service is a wrapper for the SiteMap API (see page 669): `Ektron.Cms.API.SiteMap (see page 732)`. These methods can be accessed through the SiteMap.asmx file located in:
	Tags (see page 1327)	This section describes the Web Service methods used with Ektron's Tags feature. This feature is a part Ektron's Community (see page 750) Platform. The Tags Web service is a wrapper for the Tags API (see page 669): `Ektron.Cms.API.Community.Tags (see page 805)`. These methods can be accessed through the Tags.asmx file located in:
	Taxonomy (see page 1351)	This section describes the Web Service methods used with Ektron's Taxonomy. The Taxonomy Web service is a wrapper for the Taxonomy API (see page 669): `Ektron.Cms.API.Content.Taxonomy (see page 876)`. These methods can be accessed through the Taxonomy.asmx file located in:
	ThreadedDiscussion (see page 1358)	This section describes the Web Service methods used with Ektron's Threaded Discussion feature. The Threaded Discussion Web service is a wrapper for the Threaded Discussion API (see page 669): `Ektron.Cms.API.Content.ThreadedDiscussion (see page 882)`. These methods can be accessed through the ThreadedDiscussion.asmx file located in:

	User (see page 1369)	This section describes the Web Service methods used with Users (see page 1076). The Users (see page 1076) Web service is a wrapper for the User API (see page 669): `Ektron.Cms.API.User.User` (see page 958). These methods can be accessed through the User.asmx file located in:

Ektron.Services.Task Namespace

This is namespace Ektron.Services.Task.

Classes

	Name	Description
	Task (see page 1085)	Summary description for Task
	TaskCategory (see page 1096)	Summary description for TaskCategory
	TaskCategoryType (see page 1099)	Summary description for TaskCategoryType

Classes

The following table lists classes in this documentation.

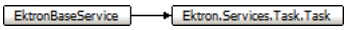
Classes

	Name	Description
	Task (see page 1085)	Summary description for Task
	TaskCategory (see page 1096)	Summary description for TaskCategory
	TaskCategoryType (see page 1099)	Summary description for TaskCategoryType

Task Class

Summary description for Task

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Task : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Task
Inherits EktronBaseService
```

File

Task.cs

Task Methods

The methods of the Task class are listed here.

Public Methods

	Name	Description
	AddTask (see page 1086)	Adds a task in the system.

	AssignTaskToUser (see page 1086)	This is the overview for the AssignTaskToUser method overload.
	AssignTaskToUsergroup (see page 1087)	This is the overview for the AssignTaskToUsergroup method overload.
	DeleteTask (see page 1089)	This is the overview for the DeleteTask method overload.
	DeleteTasks (see page 1090)	This is the overview for the DeleteTasks method overload.
	GetTask (see page 1091)	Get the task information.
	GetTasks (see page 1091)	Get All tasks from the system.
	GetTasksAssignedByUser (see page 1092)	This is the overview for the GetTasksAssignedByUser method overload.
	GetTasksAssignedToUser (see page 1093)	This is the overview for the GetTasksAssignedToUser method overload.
	GetTasksAssignedToUsergroup (see page 1094)	This is the overview for the GetTasksAssignedToUsergroup method overload.
	GetTaskState (see page 1095)	Get the task state.
	SetStateForTask (see page 1095)	Set the task state.
	UpdateTask (see page 1096)	Updates the task

Task.AddTask Method

Adds a task in the system.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTask", Description = "Adds a task")]
public bool AddTask(ref TaskData data);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTask", Description = "Adds a task")>
Public Function AddTask(ByRef data As TaskData) As bool
```

Parameters

Parameters	Description
ref TaskData data	TaskData

Returns

False=if successfully added otherwise True

AssignTaskToUser Method

This is the overview for the AssignTaskToUser method overload.

Overload List

	Name	Description
	Task.AssignTaskToUser (long, long) (see page 1087)	Assign a task to a user.
	Task.AssignTaskToUser (long, long, long) (see page 1087)	Assign a task to a user by a specified user.

Task.AssignTaskToUser Method (long, long)

Assign a task to a user.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignTaskToUser", Description = "Assign a task to a user.")]
public bool AssignTaskToUser(long taskId, long assignToUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignTaskToUser", Description = "Assign a task to a user.")>
Public Function AssignTaskToUser(taskId As long, assignToUserId As long) As bool
```

Parameters

Parameters	Description
long taskId	Task (see page 1085) ID
long assignToUserId	User (see page 1369)'s ID

Returns

False=if successfully assigned to a user otherwise True

Task.AssignTaskToUser Method (long, long, long)

Assign a task to a user by a specified user.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignTaskToUserByUser", Description = "Assign a task to a user.")]
public bool AssignTaskToUser(long taskId, long assignToUserId, long assignByUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignTaskToUserByUser", Description = "Assign a task to a user.")>
Public Function AssignTaskToUser(taskId As long, assignToUserId As long, assignByUserId As
long) As bool
```

Parameters

Parameters	Description
long taskId	Task (see page 1085) ID
long assignToUserId	Assign to User (see page 1369)'s ID
long assignByUserId	Assign by User (see page 1369)'s ID

Returns

False=if successfully assigned to a user otherwise True

AssignTaskToUsergroup Method

This is the overview for the AssignTaskToUsergroup method overload.

Overload List

	Name	Description
	Task.AssignTaskToUsergroup (long, long) (see page 1088)	Assign a task to a usergroup.
	Task.AssignTaskToUsergroup (long, long, long) (see page 1088)	Assign a task to a usergroup by a specified user.

Task.AssignTaskToUsergroup Method (long, long)

Assign a task to a usergroup.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignTaskToUsergroup", Description = "Assign a task to a
usergroup.")]
public bool AssignTaskToUsergroup(long taskId, long assignToUsergroup);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignTaskToUsergroup", Description = "Assign a task to a
usergroup.")>
Public Function AssignTaskToUsergroup(taskId As long, assignToUsergroup As long) As bool
```

Parameters

Parameters	Description
long taskId	Task (see page 1085) ID
long assignToUsergroup	Assign to Usergroup's ID

Returns

False=if successfully assigned to a usergroup otherwise True

Task.AssignTaskToUsergroup Method (long, long, long)

Assign a task to a usergroup by a specified user.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignTaskToUsergroupByUser", Description = "Assign a task to a
usergroup by a specified user.")]
public bool AssignTaskToUsergroup(long taskId, long assignToUsergroup, long assignByUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignTaskToUsergroupByUser", Description = "Assign a task to a
usergroup by a specified user.")>
Public Function AssignTaskToUsergroup(taskId As long, assignToUsergroup As long,
assignByUserId As long) As bool
```

Parameters

Parameters	Description
long taskId	Task (see page 1085) ID

long assignToUsergroup	Assign to Usergroup's ID
long assignByUserId	Assign by User (see page 1369)'s ID

Returns

False=if successfully assigned to a usergroup otherwise True

DeleteTask Method

This is the overview for the DeleteTask method overload.

Overload List

	Name	Description
	Task.DeleteTask (long) (see page 1089)	Mark the task deleted.
	Task.DeleteTask (long, bool) (see page 1089)	Mark the task deleted or delete the task permanently.

Task.DeleteTask Method (long)

Mark the task deleted.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTask", Description = "Mark the task deleted.")]
public bool DeleteTask(long taskId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTask", Description = "Mark the task deleted.")>
Public Function DeleteTask(taskId As long) As bool
```

Parameters

Parameters	Description
long taskId	Task (see page 1085) ID

Returns

False=if successfully deleted otherwise True

Task.DeleteTask Method (long, bool)

Mark the task deleted or delete the task permanently.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTaskPermanently", Description = "Mark the task deleted or
delete the task permanently.")]
public bool DeleteTask(long taskId, bool permanently);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTaskPermanently", Description = "Mark the task deleted or
delete the task permanently.")>
Public Function DeleteTask(taskId As long, permanently As bool) As bool
```

Parameters

Parameters	Description
long taskId	Task (see page 1085) ID
Permanently	True=Delete the task

Returns

False=if successfully deleted otherwise True

DeleteTasks Method

This is the overview for the DeleteTasks method overload.

Overload List

	Name	Description
	Task.DeleteTasks (long[]) (see page 1090)	Mark tasks deleted.
	Task.DeleteTasks (long[], bool) (see page 1090)	Mark tasks deleted or delete tasks permanently.

Task.DeleteTasks Method (long[])

Mark tasks deleted.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTasks", Description = "Mark tasks deleted.")]
public bool DeleteTasks(long[] taskId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTasks", Description = "Mark tasks deleted.")>
Public Function DeleteTasks(taskId As long[]) As bool
```

Parameters

Parameters	Description
long[] taskId	Array of Task (see page 1085) ID

Returns

False=if successfully deleted otherwise True

Remarks

When an error occurs the method will quite without deleting rest of the tasks.

Task.DeleteTasks Method (long[], bool)

Mark tasks deleted or delete tasks permanently.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTasksPermanently", Description = "Mark tasks deleted or delete
```

```
tasks permanently.")]
public bool DeleteTasks(long[] taskId, bool permanently);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTasksPermanently", Description = "Mark tasks deleted or delete
tasks permanently.")>
Public Function DeleteTasks(taskId As long[], permanently As bool) As bool
```

Parameters

Parameters	Description
long[] taskId	Array of Task (see page 1085) ID
Permanently	True=Delete tasks

Returns

False=if successfully deleted otherwise True

Remarks

When an error occurs the method will quite without deleting rest of the tasks.

Task.GetTask Method

Get the task information.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTask", Description = "Get the task information.")]
public TaskData GetTask(long taskId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTask", Description = "Get the task information.")>
Public Function GetTask(taskId As long) As TaskData
```

Parameters

Parameters	Description
long taskId	Task (see page 1085) Id

Returns

Task (see page 1085) Data

Task.GetTasks Method

Get All tasks from the system.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasks", Description = "")]
public TaskData[] GetTasks();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasks", Description = "")>
Public Function GetTasks() As TaskData()
```

Returns

Array of task data.

GetTasksAssignedByUser Method

This is the overview for the GetTasksAssignedByUser method overload.

Overload List

	Name	Description
	Task.GetTasksAssignedByUser (long) (see page 1092)	Get all tasks assigned by the user.
	Task.GetTasksAssignedByUser (long, Ektron.Cms.Common.EkEnumeration.TaskState) (see page 1092)	Get all task assign by the user by a state of tasks specified.

Task.GetTasksAssignedByUser Method (long)

Get all tasks assigned by the user.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedByUser", Description = "Get all tasks assigned by
the user.")]
public TaskData[] GetTasksAssignedByUser(long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedByUser", Description = "Get all tasks assigned by
the user.")>
Public Function GetTasksAssignedByUser(userId As long) As TaskData()
```

Parameters

Parameters	Description
long userId	User (see page 1369)'s ID

Returns

Array of tasks.

Task.GetTasksAssignedByUser Method (long, Ektron.Cms.Common.EkEnumeration.TaskState)

Get all task assign by the user by a state of tasks specified.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedByUserByState", Description = "")]
public TaskData[] GetTasksAssignedByUser(long userId,
Ektron.Cms.Common.EkEnumeration.TaskState taskState);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedByUserByState", Description = "")>
```

```
Public Function GetTasksAssignedByUser(userId As long, taskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As TaskData()
```

Parameters

Parameters	Description
long userId	User (see page 1369)'s ID
Ektron.Cms.Common.EkEnumeration.TaskState taskState	Task (see page 1085) State

Returns

Array of tasks.

GetTasksAssignedToUser Method

This is the overview for the GetTasksAssignedToUser method overload.

Overload List

	Name	Description
	Task.GetTasksAssignedToUser (long) (see page 1093)	Get all tasks assigned to the user.
	Task.GetTasksAssignedToUser (long, Ektron.Cms.Common.EkEnumeration.TaskState) (see page 1093)	Get all tasks assigned to the user by a state of the task specified.

Task.GetTasksAssignedToUser Method (long)

Get all tasks assigned to the user.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUser", Description = "Get all tasks assigned to
the user.")]
public TaskData[] GetTasksAssignedToUser(long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUser", Description = "Get all tasks assigned to
the user.")>
Public Function GetTasksAssignedToUser(userId As long) As TaskData()
```

Parameters

Parameters	Description
long userId	User (see page 1369)'s ID

Returns

Array of tasks.

Task.GetTasksAssignedToUser Method (long, Ektron.Cms.Common.EkEnumeration.TaskState)

Get all tasks assigned to the user by a state of the task specified.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "GetTasksAssignedToUserByState", Description = "Get all tasks
assigned to the user by a state of the task specified.")]
public TaskData[] GetTasksAssignedToUser(long userId,
Ektron.Cms.Common.EkEnumeration.TaskState taskState);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUserByState", Description = "Get all tasks
assigned to the user by a state of the task specified.")>
Public Function GetTasksAssignedToUser(userId As long, taskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As TaskData()
```

Parameters

Parameters	Description
long userId	User (see page 1369)'s ID
Ektron.Cms.Common.EkEnumeration.TaskState taskState	Task (see page 1085) State

Returns

Array of tasks.

GetTasksAssignedToUsergroup Method

This is the overview for the GetTasksAssignedToUsergroup method overload.

Overload List

	Name	Description
	Task.GetTasksAssignedToUsergroup (long) (see page 1094)	Get all tasks assigned to the usergroup.
	Task.GetTasksAssignedToUsergroup (long, Ektron.Cms.Common.EkEnumeration.TaskState) (see page 1095)	Get all tasks assigned to the usergroup by a state of tasks specified.

Task.GetTasksAssignedToUsergroup Method (long)

Get all tasks assigned to the usergroup.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUsergroup", Description = "Get all tasks assigned
to the usergroup.")]
public TaskData[] GetTasksAssignedToUsergroup(long groupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUsergroup", Description = "Get all tasks assigned
to the usergroup.")>
Public Function GetTasksAssignedToUsergroup(groupId As long) As TaskData()
```

Parameters

Parameters	Description
long groupId	Usergroup's ID

Returns

Array of tasks.

Task.GetTasksAssignedToUsergroup Method (long, Ektron.Cms.Common.EkEnumeration.TaskState)

Get all tasks assigned to the usergroup by a state of tasks specified.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUsergroupByState", Description = "Get all tasks
assigned to the usergroup by a state of tasks specified.")]
public TaskData[] GetTasksAssignedToUsergroup(long groupId,
Ektron.Cms.Common.EkEnumeration.TaskState taskState);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUsergroupByState", Description = "Get all tasks
assigned to the usergroup by a state of tasks specified.")>
Public Function GetTasksAssignedToUsergroup(groupId As long, taskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As TaskData()
```

Parameters

Parameters	Description
long groupId	Usergroup's ID
Ektron.Cms.Common.EkEnumeration.TaskState taskState	Task (see page 1085) State

Returns

Array of tasks.

Task.GetTaskState Method

Get the task state.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaskState", Description = "Get the task state.")]
public Ektron.Cms.Common.EkEnumeration.TaskState GetTaskState(long taskId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTaskState", Description = "Get the task state.")>
Public Function GetTaskState(taskId As long) As Ektron.Cms.Common.EkEnumeration.TaskState
```

Parameters

Parameters	Description
long taskId	Task (see page 1085) ID

Returns

If successful returns task state otherwise it will throw an error.

Task.SetStateForTask Method

Set the task state.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SetStateForTask", Description = "Set the task state.")]
public bool SetStateForTask(long taskId, Ektron.Cms.Common.EkEnumeration.TaskState
newTaskState);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SetStateForTask", Description = "Set the task state.")>
Public Function SetStateForTask(taskId As long, newTaskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As bool
```

Parameters

Parameters	Description
long taskId	Task (see page 1085) ID
Ektron.Cms.Common.EkEnumeration.TaskState newTaskState	New Task (see page 1085) State

Returns

False=if successfully state of the changed otherwise True

Task.UpdateTask Method

Updates the task

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateTask", Description = "Updates the task")]
public bool UpdateTask(ref TaskData data);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateTask", Description = "Updates the task")>
Public Function UpdateTask(ByRef data As TaskData) As bool
```

Parameters

Parameters	Description
ref TaskData data	Task (see page 1085) Data

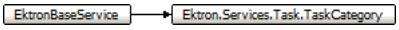
Returns

False=if successfully updated otherwise True

TaskCategory Class

Summary description for TaskCategory

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class TaskCategory : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class TaskCategory
Inherits EktronBaseService
```

File

TaskCategory.cs

TaskCategory Methods

The methods of the TaskCategory class are listed here.

Public Methods

	Name	Description
	AddTaskCategory (see page 1097)	Adds a task category.
	DeleteTaskCategory (see page 1097)	Deletes the task category.
	GetAllCategories (see page 1098)	Gets All Task (see page 1085) categories and task types belongs to those categories.
	GetTaskCategory (see page 1098)	Gets the specified task category and task types belongs to the category.
	UpdateTaskCategory (see page 1099)	Updates the category title.

TaskCategory.AddTaskCategory Method

Adds a task category.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTaskCategory", Description = "Adds a task category.")]
public long AddTaskCategory(string title);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTaskCategory", Description = "Adds a task category.")>
Public Function AddTaskCategory(title As string) As long
```

Parameters

Parameters	Description
string title	Category Title

Returns

Category Id

TaskCategory.DeleteTaskCategory Method

Deletes the task category.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTaskCategory", Description = "Deletes the task category.")]
public bool DeleteTaskCategory(long categoryId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTaskCategory", Description = "Deletes the task category.")>
Public Function DeleteTaskCategory(categoryId As long) As bool
```

Parameters

Parameters	Description
long categoryId	category id

Returns

False=if successfully deleted otherwise True

TaskCategory.GetAllCategories Method

Gets All Task (see page 1085) categories and task types belongs to those categories.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllCategories", Description = "Gets All Task categories and task
types belongs to those categories.")]
public TaskCategoryData[] GetAllCategories();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllCategories", Description = "Gets All Task categories and task
types belongs to those categories.")>
Public Function GetAllCategories() As TaskCategoryData()
```

Returns

Array of TaskCategoryData

TaskCategory.GetTaskCategory Method

Gets the specified task category and task types belongs to the category.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaskCategory", Description = "Gets the specified task category
and task types belongs to the category.")]
public TaskCategoryData GetTaskCategory(long categoryID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTaskCategory", Description = "Gets the specified task category
and task types belongs to the category.")>
Public Function GetTaskCategory(categoryID As long) As TaskCategoryData
```

Parameters

Parameters	Description
long categoryID	category id

Returns

TaskCategoryData

TaskCategory.UpdateTaskCategory Method

Updates the category title.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateTaskCategory", Description = "Updates the category title.")]
public bool UpdateTaskCategory(long categoryId, string title);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateTaskCategory", Description = "Updates the category title.")>
Public Function UpdateTaskCategory(categoryId As long, title As string) As bool
```

Parameters

Parameters	Description
long categoryId	Category Id
string title	New Title

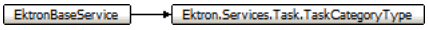
Returns

False=if successfully updated otherwise True

TaskCategoryType Class

Summary description for TaskCategoryType

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class TaskCategoryType : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class TaskCategoryType
Inherits EktronBaseService
```

File

TaskCategoryType.cs

TaskCategoryType Methods

The methods of the TaskCategoryType class are listed here.

Public Methods

	Name	Description
	AddTaskType (see page 1100)	Adds a task type to a category.
	DeleteTaskType (see page 1100)	Deletes the task type.
	GetTaskType (see page 1100)	Gets the task type information.

	UpdateTaskType (see page 1101)	Updates the task type.

TaskCategoryType.AddTaskType Method

Adds a task type to a category.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTaskType", Description = "Adds a task type to a category.")]
public long AddTaskType(string title, string description, long categoryId, bool active);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTaskType", Description = "Adds a task type to a category.")>
Public Function AddTaskType(title As string, description As string, categoryId As long, active
As bool) As long
```

Parameters

Parameters	Description
string title	Title
string description	Description
long categoryId	Task (see page 1085) Category ID
bool active	Set as active

Returns

Task (see page 1085) Type ID

TaskCategoryType.DeleteTaskType Method

Deletes the task type.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTaskType", Description = "Deletes the task type.")]
public bool DeleteTaskType(long taskTypeId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTaskType", Description = "Deletes the task type.")>
Public Function DeleteTaskType(taskTypeId As long) As bool
```

Parameters

Parameters	Description
long taskTypeId	task type id to be deleted.

Returns

False=if successfully deleted otherwise True

TaskCategoryType.GetTaskType Method

Gets the task type information.

C#

```
[SoapHeader("AuthHeader")]
```

```
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaskType", Description = "Gets a task type")]
public TaskTypeData GetTaskType(long taskTypeId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTaskType", Description = "Gets a task type")>
Public Function GetTaskType(taskTypeId As long) As TaskTypeData
```

Parameters

Parameters	Description
long taskTypeId	TaskTypeId

Returns

TaskTypeData

TaskCategoryType.UpdateTaskType Method

Updates the task type.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateTaskType", Description = "Updates the task type.")]
public bool UpdateTaskType(long taskTypeId, string title, string description, bool active);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateTaskType", Description = "Updates the task type.")>
Public Function UpdateTaskType(taskTypeId As long, title As string, description As string,
active As bool) As bool
```

Parameters

Parameters	Description
long taskTypeId	Task (see page 1085) Type Id
string title	Title
string description	Description
bool active	Set as active

Returns

False=if successfully updated otherwise True

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	ActiveDirectoryUser (see page 1103)	This section describes the Web Service methods used with Ektron's Active Directory (see page 196). The Active Directory (see page 196) Web service is a wrapper for the Active Directory (see page 196) API (see page 669): `Ektron.Cms.API.User.ADUser (see page 951)`. These methods can be accessed through the ActiveDirectoryUser.asmx file located in:
	Asset (see page 1112)	This section describes the Web Service methods used with Assets. The Assets Web service is a wrapper for the Asset API (see page 669): `Ektron.Cms.API.Content.Asset (see page 822)`. These methods can be accessed through the Asset.asmx file located in:

	Blog (see page 1116)	This section describes the Web Service methods used with blogs. The Blog Web service is a wrapper for the blog API (see page 669): `Ektron.Cms.API.Content.Blogs.` These methods can be accessed through the blog.asmx file located in:
	BusinessRules (see page 1130)	This section describes the Web Service methods used with Ektron's Business Rules. The Business Rules Web service is a wrapper for the Business Rules API (see page 669): `Ektron.Cms.API.BusinessRule (see page 671)`. These methods can be accessed through the BusinessRules.asmx file located in:
	Calendar (see page 1133)	Summary description for Calendar
	Analytics (see page 1133)	This section describes the Web Service methods used with analytics. The Analytics Web service is a wrapper for the Analytics API (see page 669): `Ektron.Cms.API.Analytics.Analytics (see page 745)`. These methods can be accessed through the Analytics.asmx file located in:
	CommunityGroup (see page 1181)	This section describes the Web Service methods used with Ektron's Community (see page 750) Groups. The Community (see page 750) Groups Web service is a wrapper for the Community (see page 750) Groups API (see page 669): `Ektron.Cms.API.Community.CommunityGroup (see page 750)`. These methods can be accessed through the CommunityGroup.asmx file located in:
	Content (see page 1198)	This section describes the Web Service methods used with content. The Content Web service is a wrapper for the Content API (see page 669): `Ektron.Cms.API.Content.Content (see page 840)`. These methods can be accessed through the Content.asmx file located in:
	ContentRating (see page 1214)	This section describes the Web Service methods used with Ektron's Content (see page 1198) Ratings. The Content (see page 1198) Rating Web service is a wrapper for the Content (see page 1198) Rating API (see page 669): `Ektron.Cms.API.Content.ContentRating (see page 866)`. These methods can be accessed through the ContentRating.asmx file located in:
	CustomFields (see page 1216)	This section describes the Web Service methods used with Ektron's Custom Fields. The Custom Fields Web service is a wrapper for the Custom Fields API (see page 669): `Ektron.Cms.API.CustomFields (see page 673)`. These methods can be accessed through the CustomFields.asmx file located in:
	Favorites (see page 1225)	This section describes the Web Service methods used with Ektron's Favorites feature. This feature is a part Ektron's Community (see page 750) Platform. The Favorites Web service is a wrapper for the Favorites API (see page 669): `Ektron.Cms.API.Community.Favorites (see page 763)`. These methods can be accessed through the Favorites.asmx file located in:
	Flag (see page 1231)	This section describes the Web Service methods used with Ektron's Flagging (see page 771) feature. This feature is a part Ektron's Community (see page 750) Platform. The Flagging (see page 771) Web service is a wrapper for the Flagging (see page 771) API (see page 669): `Ektron.Cms.API.Community.Flagging (see page 771)`. These methods can be accessed through the Flag.asmx file located in:
	Folder (see page 1236)	This section describes the Web Service methods used with folder. The Folder Web service is a wrapper for the Folder API (see page 669): `Ektron.Cms.API.Folder (see page 683)`. These methods can be accessed through the Folder.asmx file located in:
	Font (see page 1244)	This section describes the Web Service methods used with Fonts. The Font Web service is a wrapper for the Font API (see page 669): `Ektron.Cms.API.Font`. These methods can be accessed through the Font.asmx file located in:
	Form (see page 1247)	This section describes the Web Service methods used with forms. The Form Web service is a wrapper for the Form API (see page 669): `Ektron.Cms.API.Content.Form (see page 871)`. These methods can be accessed through the form.asmx file located in:
	Friends (see page 1255)	This section describes the Web Service methods used with Ektron's Friends feature. This feature is a part Ektron's Community (see page 750) Platform. The Friends Web service is a wrapper for the Friends API (see page 669): `Ektron.Cms.API.Community.Friends (see page 777)`. These methods can be accessed through the Friends.asmx file located in:
	Library (see page 1269)	This section describes the Web Service methods used with the Library. The Library Web service is a wrapper for the Library API (see page 669): `Ektron.Cms.API.Library (see page 695)`. These methods can be accessed through the Library.asmx file located in:
	MessageBoard (see page 1279)	This section describes the Web Service methods used with Ektron's Message Board feature. This feature is a part Ektron's Community (see page 750) Platform. The MessageBoard Web service is a wrapper for the MessageBoard API (see page 669): `Ektron.Cms.API.Community.MessageBoard (see page 787)`. These methods can be accessed through the MessageBoard.asmx file located in:
	Metadata (see page 1300)	This section describes the Web Service methods used with Metadata. The Metadata Web service is a wrapper for the Metadata API (see page 669): `Ektron.Cms.API.Metadata (see page 714)`. These methods can be accessed through the Metadata.asmx file located in:
	Permissions (see page 1304)	This section describes the Web Service methods used with Permissions. The Permissions Web service is a wrapper for the Permissions API (see page 669): `Ektron.Cms.API.Permissions (see page 721)`. These methods can be accessed through the Permissions.asmx file located in:
	SearchManager (see page 1311)	This section describes the Web Service methods used with the SearchManager. The SearchManager Web service is a wrapper for the SearchManager API (see page 669): `Ektron.Cms.API.Search.SearchManager (see page 899)`. These methods can be accessed through the SearchManager.asmx file located in:
	Site (see page 1316)	This section describes the Web Service methods used with Site functionality. The Site Web service is a wrapper for the Site API (see page 669): `Ektron.Cms.API.Site (see page 725)`. These methods can be accessed through the Site.asmx file located in:
	SiteMap (see page 1324)	This section describes the Web Service methods used with SiteMap. The SiteMap Web service is a wrapper for the SiteMap API (see page 669): `Ektron.Cms.API.SiteMap (see page 732)`. These methods can be accessed through the SiteMap.asmx file located in:

	Tags (see page 1327)	This section describes the Web Service methods used with Ektron's Tags feature. This feature is a part Ektron's Community (see page 750) Platform. The Tags Web service is a wrapper for the Tags API (see page 669): `Ektron.Cms.API.Community.Tags` (see page 805). These methods can be accessed through the Tags.asmx file located in:
	Taxonomy (see page 1351)	This section describes the Web Service methods used with Ektron's Taxonomy. The Taxonomy Web service is a wrapper for the Taxonomy API (see page 669): `Ektron.Cms.API.Content.Taxonomy` (see page 876). These methods can be accessed through the Taxonomy.asmx file located in:
	ThreadedDiscussion (see page 1358)	This section describes the Web Service methods used with Ektron's Threaded Discussion feature. The Threaded Discussion Web service is a wrapper for the Threaded Discussion API (see page 669): `Ektron.Cms.API.Content.ThreadedDiscussion` (see page 882). These methods can be accessed through the ThreadedDiscussion.asmx file located in:
	User (see page 1369)	This section describes the Web Service methods used with Users (see page 1076). The Users (see page 1076) Web service is a wrapper for the User API (see page 669): `Ektron.Cms.API.User.User` (see page 958). These methods can be accessed through the User.asmx file located in:

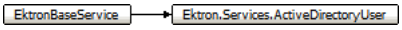
ActiveDirectoryUser Class

This section describes the Web Service methods used with Ektron's Active Directory (see page 196). The Active Directory (see page 196) Web service is a wrapper for the Active Directory (see page 196) API (see page 669): `Ektron.Cms.API.User.ADUser` (see page 951). These methods can be accessed through the ActiveDirectoryUser.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class ActiveDirectoryUser : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class ActiveDirectoryUser
Inherits EktronBaseService
```

File

ActiveDirectoryUser.cs

ActiveDirectoryUser Methods

The methods of the ActiveDirectoryUser class are listed here.

Public Methods

	Name	Description
	AddADGroupToCMS (see page 1104)	This function allows you add Active Directory (see page 196) groups to the CMS. This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	DeSynchUserGroups (see page 1104)	Updates the group with a new group name. This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	DeSynchUsers (see page 1105)	Update the user with new user details. This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	GetADMapping (see page 1105)	Loads the CMS and Active Directory (see page 196) name. This method can be accessed through the ActiveDirectoryUser.asmx file located in:

	GetADStatus (see page 1106)	Used to load required Synch, Desynch and relationship flags of the configured Active Directory (see page 196). This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	GetAvailableADGroups (see page 1106)	Loads all of the Active Directory (see page 196) groups by a given domain and filter. This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	GetAvailableADUsers (see page 1107)	Loads all available Active Directory (see page 196) users. This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	GetCMSGroupsToSync (see page 1108)	Loads all the CMS groups that need to be synchronized. This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	GetCMSRelationshipsToSync (see page 1108)	Loads all the CMS relationships that need to be synchronized with Active Directory (see page 196). This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	GetCMSUsersToSync (see page 1109)	Loads all the CMS users that need to be synchronized with Active Directory (see page 196). This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	GetDomains (see page 1109)	Loads a list of domains from Active Directory (see page 196). This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	MapCMSUserGroupToAD (see page 1110)	Updates the given CMS group corresponding to the Active Directory (see page 196) group. This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	MapCMSUserToAD (see page 1110)	Maps the given user ID to the corresponding Active Directory (see page 196) entry. This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	UpdateADMapping (see page 1111)	Updates the Active directory information into the CMS based on the user ID and user data. This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	UpdateADUsersGroups (see page 1111)	Updates Active Directory (see page 196) users in a CMS group. This method can be accessed through the ActiveDirectoryUser.asmx file located in:
	UpdateUsersInfoFromAD (see page 1112)	Updates the user details from the Active Directory (see page 196) details. This method can be accessed through the ActiveDirectoryUser.asmx file located in:

ActiveDirectoryUser.AddADGroupToCMS Method

This function allows you add Active Directory (see page 196) groups to the CMS.

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddADGroupToCMS", Description = "")]
public void AddADGroupToCMS(object GroupPaths);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddADGroupToCMS", Description = "")>
Public Function AddADGroupToCMS(GroupPaths As object) As void
```

Parameters

Parameters	Description
object GroupPaths	A collection of Active Directory (see page 196) group paths.

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.DeSynchUserGroups Method

Updates the group with a new group name.

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeSynchUserGroups", Description = "")]
public AdDeSyncGroupData[] DeSynchUserGroups(bool Flag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeSynchUserGroups", Description = "")>
Public Function DeSynchUserGroups(Flag As bool) As AdDeSyncGroupData()
```

Parameters

Parameters	Description
bool Flag	If true, update the user group with the new group information, otherwise none.

Returns

Ektron.Cms.AdDeSyncGroupData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.DeSynchUsers Method

Update the user with new user details.

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeSynchUsers", Description = "")]
public AdDeSyncUserData[] DeSynchUsers(bool Flag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeSynchUsers", Description = "")>
Public Function DeSynchUsers(Flag As bool) As AdDeSyncUserData()
```

Parameters

Parameters	Description
bool Flag	If true, update the user with new user information, otherwise none.

Returns

Ektron.Cms.AdDeSyncUserData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.GetADMapping Method

Loads the CMS and Active Directory (see page 196) name.

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetADMapping", Description = "")]
public AdMappingData[] GetADMapping(long CallerId, string MapType, int Map, int Exclude, int
Editable);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetADMapping", Description = "")>
Public Function GetADMapping(CallerId As long, MapType As string, Map As Integer, Exclude As
Integer, Editable As Integer) As AdMappingData()
```

Parameters

Parameters	Description
long CallerId	Who is calling this function.
string MapType	Possible value is “userprop” (user properties).
int Map	Ektron.Cms.Common.EkEnumeration.MapTypes (include, exclude and dontconsider are the possible values).
int Exclude	Flag (see page 1231) (0 – false, 1-true).
int Editable	Flag (see page 1231)(0-false,1-true).

Returns

Ektron.Cms.AdMappingData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.GetADStatus Method

Used to load required Synch, Desynch and relationship flags of the configured Active Directory (see page 196).

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetADStatus", Description = "")]
public AdSyncData GetADStatus();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetADStatus", Description = "")>
Public Function GetADStatus() As AdSyncData
```

Returns

Ektron.Cms.AdSyncData

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.GetAvailableADGroups Method

Loads all of the Active Directory (see page 196) groups by a given domain and filter.

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAvailableADGroups", Description = "")]
public GroupData[] GetAvailableADGroups(string Filter, string Domain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAvailableADGroups", Description = "")>
Public Function GetAvailableADGroups(Filter As string, Domain As string) As GroupData()
```

Parameters

Parameters	Description
string Filter	Filter by complete or a specific group name.
string Domain	The domain name. (Required)

Returns

Ektron.Cms.GroupData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.GetAvailableADUsers Method

Loads all available Active Directory (see page 196) users.

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAvailableADUsers", Description = "")]
public UserData[] GetAvailableADUsers(object Attributes, object Filter, string OrderBy, string
Domain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAvailableADUsers", Description = "")>
Public Function GetAvailableADUsers(Attributes As object, Filter As object, OrderBy As string,
Domain As string) As UserData()
```

Parameters

Parameters	Description
object Attributes	Possible attributes are firstname, lastname, username or any(*).
object Filter	Possible filters are firstname, lastname, username or any(*).
string OrderBy	Any specific order such as username, firstname, lastname and domain.
string Domain	The active directory domain. (Required)

Returns

Ektron.Cms.UserData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.GetCMSGroupsToSync Method

Loads all the CMS groups that need to be synchronized.

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCMSGroupsToSync", Description = "")]
public GroupData[] GetCMSGroupsToSync(int MaxReturn);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCMSGroupsToSync", Description = "")>
Public Function GetCMSGroupsToSync(MaxReturn As Integer) As GroupData()
```

Parameters

Parameters	Description
int MaxReturn	The maximum number of groups that should be displayed.

Returns

Ektron.Cms.GroupData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.GetCMSRelationshipsToSync Method

Loads all the CMS relationships that need to be synchronized with Active Directory (see page 196).

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCMSRelationshipsToSync", Description = "")]
public UserGroupData[] GetCMSRelationshipsToSync(int MaxReturn);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCMSRelationshipsToSync", Description = "")>
Public Function GetCMSRelationshipsToSync(MaxReturn As Integer) As UserGroupData()
```

Parameters

Parameters	Description
int MaxReturn	The maximum number of results that should be displayed.

Returns

Ektron.Cms.UserGroupData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.GetCMSUsersToSync Method

Loads all the CMS users that need to be synchronized with Active Directory (see page 196).

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCMSUsersToSync", Description = "")]
public UserData[] GetCMSUsersToSync(int MaxReturn);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCMSUsersToSync", Description = "")>
Public Function GetCMSUsersToSync(MaxReturn As Integer) As UserData()
```

Parameters

Parameters	Description
int MaxReturn	The maximum number of users that should be displayed.

Returns

Ektron.Cms.UserData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.GetDomains Method

Loads a list of domains from Active Directory (see page 196).

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDomains", Description = "")]
public DomainData[] GetDomains(int ShowAll, int CheckIntegration);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDomains", Description = "")>
Public Function GetDomains(ShowAll As Integer, CheckIntegration As Integer) As DomainData()
```

Parameters

Parameters	Description
int ShowAll	flag(1 = show all).
int CheckIntegration	Check whether CMS is configured to integrate Active Directory (see page 196) (1 = check otherwise resumed).

Returns

Ektron.Cms.DomainData()

ActiveDirectoryUser.MapCMSUserGroupToAD Method

Updates the given CMS group corresponding to the Active Directory (see page 196) group.

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MapCMSUserGroupToAD", Description = "")]
public void MapCMSUserGroupToAD(long GroupId, string GroupName, string Domain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MapCMSUserGroupToAD", Description = "")>
Public Function MapCMSUserGroupToAD(GroupId As long, GroupName As string, Domain As string) As
void
```

Parameters

Parameters	Description
long GroupId	CMS group ID.
string GroupName	CMS group name.
string Domain	The Active Directory (see page 196) domain.

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.MapCMSUserToAD Method

Maps the given user ID to the corresponding Active Directory (see page 196) entry.

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MapCMSUserToAD", Description = "")]
public void MapCMSUserToAD(long UserId, string UserName, string Domain, int PullInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MapCMSUserToAD", Description = "")>
Public Function MapCMSUserToAD(UserId As long, UserName As string, Domain As string, PullInfo
As Integer) As void
```

Parameters

Parameters	Description
long UserId	The user's ID.
string UserName	The user's name.
string Domain	The user's domain.
int PullInfo	Flag (see page 1231) to load Active Directory (see page 196) information. (1 = Load Active Directory (see page 196) information, otherwise none.)

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.UpdateADMapping Method

Updates the Active directory information into the CMS based on the user ID and user data.

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateADMapping", Description = "")]
public void UpdateADMapping(long UserId, object pagedata);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateADMapping", Description = "")>
Public Function UpdateADMapping(UserId As long, pagedata As object) As void
```

Parameters

Parameters	Description
long UserId	The user's ID.
object pagedata	Key, value pair for given user ID.

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.UpdateADUsersGroups Method

Updates Active Directory (see page 196) users in a CMS group.

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateADUsersGroups", Description = "")]
public void UpdateADUsersGroups(string UserName, string Domain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateADUsersGroups", Description = "")>
Public Function UpdateADUsersGroups(UserName As string, Domain As string) As void
```

Parameters

Parameters	Description
string UserName	The user's username.
string Domain	The user's Domain.

Remarks

A user with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.UpdateUsersInfoFromAD Method

Updates the user details from the Active Directory (see page 196) details.

This method can be accessed through the ActiveDirectoryUser.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/ActiveDirectoryUser.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateUsersInfoFromAD", Description = "")]
public void UpdateUsersInfoFromAD(string UserName, string Domain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateUsersInfoFromAD", Description = "")>
Public Function UpdateUsersInfoFromAD(UserName As string, Domain As string) As void
```

Parameters

Parameters	Description
string UserName	The user's username.
string Domain	The user's Domain.

Remarks

A user with administrator privileges must be logged in to perform this operation.

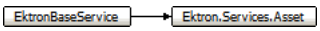
Asset Class

This section describes the Web Service methods used with Assets. The Assets Web service is a wrapper for the Asset API (see page 669): `Ektron.Cms.API.Content.Asset` (see page 822). These methods can be accessed through the Asset.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Asset.asmx
```

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class Asset : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class Asset
Inherits EktronBaseService
```

File

Asset.cs

Asset Methods

The methods of the Asset class are listed here.

Public Methods

	Name	Description
	GetAssetMgtConfigInfo (see page 1113)	Loads all of the registered plug-in configuration details. This method can be accessed through the Asset.asmx file located in:
	GetAssetMgtInfo (see page 1113)	Loads all of the registered plug-in type details. This method can be accessed through the Asset.asmx file located in:
	GetAssetSupertypes (see page 1114)	Loads all of the asset supertypes supported by documents add-in. This method can be accessed through the Asset.asmx file located in:
	GetEditSnippet (see page 1114)	Loads the edit snippet for the given document. Used with the Document Management functionality. This method can be accessed through the Asset.asmx file located in:
	GetMyAssetMgtID (see page 1115)	Retrieves the Document Management functionality client ID. This method can be accessed through the Asset.asmx file located in:
	MakeCopyDMSAsset (see page 1115)	Create a copy of DMS Asset (see page 1112). This method can be accessed through the Asset.asmx file located in:
	SetAssetMgtConfigInfo (see page 1115)	Updates the Document Management configuration information. This method can be accessed through the Asset.asmx file located in:

Asset.GetAssetMgtConfigInfo Method

Loads all of the registered plug-in configuration details.

This method can be accessed through the Asset.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Asset.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAssetMgtConfigInfo", Description = "")]
public AssetConfigInfo[] GetAssetMgtConfigInfo();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAssetMgtConfigInfo", Description = "")>
Public Function GetAssetMgtConfigInfo() As AssetConfigInfo()
```

Returns

Ektron.Cms.AssetConfigInfo()

Asset.GetAssetMgtInfo Method

Loads all of the registered plug-in type details.

This method can be accessed through the Asset.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Asset.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAssetMgtInfo", Description = "")]
public AssetInfoData[] GetAssetMgtInfo();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAssetMgtInfo", Description = "")>
Public Function GetAssetMgtInfo() As AssetInfoData()
```

Returns

Ektron.Cms.AssetInfoData()

Asset.GetAssetSupertypes Method

Loads all of the asset supertypes supported by documents add-in.

This method can be accessed through the Asset.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Asset.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAssetSupertypes", Description = "")]
public AssetInfoData[] GetAssetSupertypes();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAssetSupertypes", Description = "")>
Public Function GetAssetSupertypes() As AssetInfoData()
```

Returns

Ektron.Cms.AssetInfoData()

Asset.GetEditSnippet Method

Loads the edit snippet for the given document. Used with the Document Management functionality.

This method can be accessed through the Asset.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Asset.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetEditSnippet", Description = "")]
public string GetEditSnippet(string AssetId, int ContentType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetEditSnippet", Description = "")>
Public Function GetEditSnippet(AssetId As string, ContentType As Integer) As string
```

Parameters

Parameters	Description
string AssetId	The document's ID (Required)
int ContentType	The type of the content, such as office documents or managed files or * (all).

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetMyAssetMgtID Method

Retrieves the Document Management functionality client ID.

This method can be accessed through the Asset.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Asset.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMyAssetMgtID", Description = "")]
public string GetMyAssetMgtID();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMyAssetMgtID", Description = "")>
Public Function GetMyAssetMgtID() As string
```

Returns

String

Asset.MakeCopyDMSAsset Method

Create a copy of DMS Asset (see page 1112).

This method can be accessed through the Asset.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Asset.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MakeCopyDMSAsset", Description = "")]
public bool MakeCopyDMSAsset(string AssetID, string AssetVersion);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MakeCopyDMSAsset", Description = "")>
Public Function MakeCopyDMSAsset(AssetID As string, AssetVersion As string) As bool
```

Parameters

Parameters	Description
strAssetID	ID associated with the asset
strAssetVersion	Version of the asset

Returns

boolean

Remarks

User (see page 1369) login with create content privileges are required to perform this operation.

Asset.SetAssetMgtConfigInfo Method

Updates the Document Management configuration information.

This method can be accessed through the Asset.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Asset.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SetAssetMgtConfigInfo", Description = "")]
public bool SetAssetMgtConfigInfo(AssetConfigInfo[] AssetConfig);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SetAssetMgtConfigInfo", Description = "")>
Public Function SetAssetMgtConfigInfo(AssetConfig As AssetConfigInfo[]) As bool
```

Parameters

Parameters	Description
asset_config	The Document Management configuration data

Returns

Boolean

Remarks

User (see page 1369) login with administrator privileges are required to perform this operation.

Blog Class

This section describes the Web Service methods used with blogs. The Blog Web service is a wrapper for the blog API (see page 669): `Ektron.Cms.API.Content.Blogs.` These methods can be accessed through the blog.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx`

Web Services (see page 1083) examples involving blogs are available in the Developer sample site. You can download this site from:

http://www.ektron.com/cms400-web-cms.aspx?id=5278

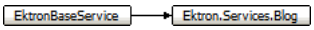
Once the developer site is installed, example code files are located in your site root. For example:

`C:\Inetpub\wwwroot\CMS400Developer\Developer\dotNetAssemblies\WebServicesAPI\ContentAPI\Blog.`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header. For example:

```
blogApi.AuthenticationHeaderValue = New localhost.AuthenticationHeader()
blogApi.AuthenticationHeaderValue.Username = "admin"
blogApi.AuthenticationHeaderValue.Password = "admin"
blogApi.AuthenticationHeaderValue.Domain = "Domain" '''Only needed if AD/LDAP is being used
```

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Blog : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Blog
Inherits EktronBaseService
```

File

Blog.cs

Blog Methods

The methods of the Blog class are listed here.

Public Methods

	Name	Description
	AddBlog (see page 1117)	Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog. This method can be accessed through the blog.asmx file located in:
	AddCommentForPost (see page 1118)	Adds a post comment to a blog post. Accepts a BlogComment class as the input. Will return the result of that add action. False indicates the action was successful. This method can be accessed through the blog.asmx file located in:
	AddPost (see page 1119)	Adds a post to a given blog. Accepts a ContentData class as the post content. Return new post ID. This method can be accessed through the blog.asmx file located in:
	DeleteBlog (see page 1119)	Deletes a blog with a given ID. This will delete the blog as well as all posts and comments for the blog. BlogID must be greater than 0. This method can be accessed through the blog.asmx file located in:
	DeleteBlogPostComment (see page 1120)	Deletes the blog post comment by it's id. This method can be accessed through the blog.asmx file located in:
	DeletePost (see page 1120)	Deletes a post by the post's ID This method will delete all comments associated with the post as well. Returns boolean value. This method can be accessed through the blog.asmx file located in:
	GetBlankBlogPostData (see page 1120)	Returns a blank BlogPostData class. Used Primarily in the Workarea. This method can be accessed through the blog.asmx file located in:
	GetBlog (see page 1121)	This is the overview for the GetBlog method overload.
	GetBlogData (see page 1123)	This is the overview for the GetBlogData method overload.
	GetBlogDataForPost (see page 1126)	Returns the blog information for a blog post. This contains information about categories, whether comments are enabled, etc. Accepts the post ID. This method can be accessed through the blog.asmx file located in:
	GetBlogPostData (see page 1127)	Gets the Post data for a specific blog post. Accepts the ID of the post. Will return tags, trackback, pingback, and categories for the post. This method can be accessed through the blog.asmx file located in:
	GetBlogRoll (see page 1127)	This is the overview for the GetBlogRoll method overload.
	GetPost (see page 1129)	Gets a blog post for a given ID. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData Class. This method can be accessed through the blog.asmx file located in:
	GetPostbyID (see page 1129)	Gets a blog post for a given ID. Includes trackback, pinback, comments, and other information. Returns BlogPostData Class. This method can be accessed through the blog.asmx file located in:
	GetUserBlog (see page 1130)	Gets a blog based on a user's ID. This method can be accessed through the blog.asmx file located in:
	UpdateBlogProperties (see page 1130)	This method updates the blog data. This method can be accessed through the blog.asmx file located in:

Blog.AddBlog Method

Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.

This method can be accessed through the blog.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddBlog", Description = "")]
public long AddBlog(long ParentFolderID, string BlogName, string BlogTitle, string
```

```
BlogDescription, Ektron.Cms.Common.EkEnumeration.BlogVisibility Visibility, bool
EnableComments, bool ModerateComments, bool RequireAuthentication, string[] Categories,
BlogRoll RollLinks);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddBlog", Description = "")>
Public Function AddBlog(ParentFolderID As long, BlogName As string, BlogTitle As string,
BlogDescription As string, Visibility As Ektron.Cms.Common.EkEnumeration.BlogVisibility,
EnableComments As bool, ModerateComments As bool, RequireAuthentication As bool, Categories As
string[], RollLinks As BlogRoll) As long
```

Parameters

Parameters	Description
long ParentFolderID	
string BlogName	Name of the Blog (see page 1116).
string BlogTitle	Title of the Blog (see page 1116).
string BlogDescription	Description of the Blog (see page 1116).
Ektron.Cms.Common.EkEnumeration.BlogVisibility Visibility	Is the blog public/private?
bool EnableComments	Allow commenting.
bool ModerateComments	If commenting is allowed, moderate comments.
bool RequireAuthentication	If commenting is allowed, require authentication to add comment.
string[] Categories	String Array representing the categories for this blog.
BlogRoll RollLinks	The Blog (see page 1116)'s roll links.

Returns

The numeric value of the blog ID that is greater than 0.

Blog.AddCommentForPost Method

Adds a post comment to a blog post. Accepts a BlogComment class as the input. Will return the result of that add action. False indicates the action was successful.

This method can be accessed through the blog.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddCommentForPost", Description = "")]
public bool AddCommentForPost(BlogComment bcComment);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddCommentForPost", Description = "")>
Public Function AddCommentForPost(bcComment As BlogComment) As bool
```

Parameters

Parameters	Description
BlogComment bcComment	A populated BlogComment class. Title, DisplayName, Comment text, Email are required.

Returns

A Boolean value indicating whether there was a problem. False indicates the comment was added successfully.

Blog.AddPost Method

Adds a post to a given blog. Accepts a ContentData class as the post content. Return new post ID.

This method can be accessed through the blog.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddPost", Description = "")]
public long AddPost(long blogID, ContentData PostContent, string[] Categories, bool Pingback,
string Tags, string TrackbackURL);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddPost", Description = "")>
Public Function AddPost(blogID As long, PostContent As ContentData, Categories As string[],
Pingback As bool, Tags As string, TrackbackURL As string) As long
```

Parameters

Parameters	Description
long blogID	Numeric ID of the blog to add the post to.
ContentData PostContent	ContentData class representing the HTML, title, etc.
string[] Categories	String Array representing the categories this post applies to.
bool Pingback	Pingback URL(s) in this post?
string Tags	String list of tags that that applies to this post.
string TrackbackURL	Trackback URL(s) to notify.

Returns

Numeric ID of the post created.

Blog.DeleteBlog Method

Deletes a blog with a given ID. This will delete the blog as well as all posts and comments for the blog. BlogID must be greater than 0.

This method can be accessed through the blog.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteBlog", Description = "")]
public void DeleteBlog(long BlogID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteBlog", Description = "")>
Public Function DeleteBlog(BlogID As long) As void
```

Parameters

Parameters	Description
long BlogID	The numeric ID of the blog. This corresponds to the folder ID of the blog.

Blog.DeleteBlogPostComment Method

Deletes the blog post comment by it's id.

This method can be accessed through the blog.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteBlogPostComment", Description = "")]
public void DeleteBlogPostComment(Int64 postComment_id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteBlogPostComment", Description = "")>
Public Function DeleteBlogPostComment(postComment_id As Int64) As void
```

Parameters

Parameters	Description
Int64 postComment_id	post comment id

Blog.DeletePost Method

Deletes a post by the post's ID This method will delete all comments associated with the post as well. Returns boolean value.

This method can be accessed through the blog.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeletePost", Description = "")]
public bool DeletePost(long post_id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeletePost", Description = "")>
Public Function DeletePost(post_id As long) As bool
```

Parameters

Parameters	Description
long post_id	The numeric value of the post ID you wish to delete.

Returns

Boolean

Blog.GetBlankBlogPostData Method

Returns a blank BlogPostData class. Used Primarily in the Workarea.

This method can be accessed through the blog.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx
```

C#

```
[SoapHeader("AuthHeader")]
```

```
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlankBlogPostData", Description = "")]
public BlogPostData GetBlankBlogPostData();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlankBlogPostData", Description = "")>
Public Function GetBlankBlogPostData() As BlogPostData
```

Returns

A blank, but not null, BlogPostData class.

GetBlog Method

This is the overview for the GetBlog method overload.

Description

Returns a BlogData class for a given blog id, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. This method is overloaded.

- GetBlog (Long, String, Int, Int) - returns a BlogData class for a given blog by providing the blog's ID, the date to start getting posts, the maximum number of entries to return and the ID of the Language.
- GetBlog (String, String, Int, Int) - returns a BlogData class for a given blog by providing the blog's folder path in CMS400.NET, the date to start getting posts, the maximum number of entries to return and the ID of the Language.

Overload List

	Name	Description
	Blog.GetBlog (long, string, int, int) (see page 1121)	Returns a BlogData class for a given blog ID, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. This method can be accessed through the blog.asmx file located in:
	Blog.GetBlog (string, string, int, int) (see page 1122)	Returns a BlogData class for a given path, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. This method can be accessed through the blog.asmx file located in:

Blog.GetBlog Method (long, string, int, int)

Returns a BlogData class for a given blog ID, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

This method can be accessed through the blog.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx`

This function is overloaded. See Also:

GetBlog (int, string, int, int) (see page 1122)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlog by ID, StartDate, MaxEntries and LanguageID", Description =
"")]
public BlogData GetBlog(long ID, string StartDate, int MaxEntries, int LanguageID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlog by ID, StartDate, MaxEntries and LanguageID", Description =
"")>
Public Function GetBlog(ID As long, StartDate As string, MaxEntries As Integer, LanguageID As
Integer) As BlogData
```

Parameters

Parameters	Description
long ID	The ID of the blog. This is a numeric value that corresponds to the folder ID of the blog.
string StartDate	The date to get entries from. Used if you want to get historical information. Gets information only for the date specified.
int MaxEntries	The total number of posts to return. If set to 0, the `postsvisible` property of the blog is used. If set to -1, the content is not returned, only the post info for the content.
int LanguageID	The language ID for the content to get.

Returns

For a given blog ID, this returns either null or a BlogData class with the associated posts. This also includes the roll, categories, and post information.

Remarks

Developers are advised to check for null prior to using.

Blog.GetBlog Method (string, string, int, int)

Returns a BlogData class for a given path, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

This method can be accessed through the blog.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx`

This function is overloaded. See Also:

GetBlog (string, string, int, int)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlog By Path, StartDate, MaxEntries and LanguageID", Description
= "")]
public BlogData GetBlog(string path, string StartDate, int MaxEntries, int LanguageID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlog By Path, StartDate, MaxEntries and LanguageID", Description
= "")>
Public Function GetBlog(path As string, StartDate As string, MaxEntries As Integer, LanguageID
As Integer) As BlogData
```

Parameters

Parameters	Description
string path	The path of the blog. This is a string value that corresponds to the folder path of the blog.
string StartDate	The date to get entries from. Used if you want to get historical information. Gets information only for the date specified.

int MaxEntries	The total number of posts to return. If set to 0, the `postsvisible` property of the blog is used. If set to -1, the content is not returned, only the post info for the content.
int LanguageID	The language ID for the content to get.

Returns

For a given path, this returns either null or a BlogData class with the associated posts. This also includes the roll, categories, and post information.

Remarks

Developers are advised to check for null prior to using.

GetBlogData Method

This is the overview for the GetBlogData method overload.

Description

Overloaded Method to obtain a BlogData Class.

- GetBlogData (FolderData) - returns the corresponding blog (if applicable) for a complete folderData object.
- GetBlogData (Long) - returns either a populated BlogData class or null (nothing) by providing the ID of the blog.
- GetBlogData (Long, DateTime)- returns either a populated BlogData class or null (nothing) by providing the ID of the blog and date to start returning data.
- GetBlogData (Long, DateTime, Int, Int)- returns either a populated BlogData class or null (nothing) by providing the integer ID of the blog, the date to start returning data, the number of posts to return and the language ID.

Overload List

	Name	Description
	Blog.GetBlogData (FolderData) (see page 1123)	Returns the corresponding blog (if applicable) for a complete folderData object. If there is code that already leverages our FolderData class, this can piggyback atop it. **Returns only if the folder is a blog.** This method can be accessed through the blog.asmx file located in:
	Blog.GetBlogData (long) (see page 1124)	Overloaded Method to obtain a BlogData Class. Accepts the Numeric ID of the blog. Returns either a populated BlogData class or null (nothing). This method can be accessed through the blog.asmx file located in:
	Blog.GetBlogData (long, DateTime) (see page 1125)	Overloaded Method to obtain a BlogData Class. Accepts the numeric ID of the blog, as well as the StartDate (as DateTime) for the posts. Returns either a populated BlogData class or null (nothing). This method can be accessed through the blog.asmx file located in:
	Blog.GetBlogData (long, DateTime, int, int) (see page 1125)	Overloaded Method to obtain a BlogData Class. Accepts the numeric ID of the blog, the StartDate (as DateTime) for the posts, the number of posts visible and the language ID. Returns either a populated BlogData class or null (nothing). This method can be accessed through the blog.asmx file located in:

Blog.GetBlogData Method (FolderData)

Returns the corresponding blog (if applicable) for a complete folderData object. If there is code that already leverages our FolderData class, this can piggyback atop it. **Returns only if the folder is a blog.**

This method can be accessed through the blog.asmx file located in:

~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx

This function is overloaded. See Also:

GetBlogData (long) (see page 1124)

GetBlogData (long, DateTime) (see page 1125)

GetBlogData (long, DateTime, int, int) (see page 1125)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogData By FolderData", Description = "")]
public BlogData GetBlogData(FolderData folder_data);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogData By FolderData", Description = "")>
Public Function GetBlogData(folder_data As FolderData) As BlogData
```

Parameters

Parameters	Description
FolderData folder_data	A populated FolderData class. (see FolderData)

Returns

A populated BlogData class or nothing

Remarks

Blog.GetBlogData Method (long)

Overloaded Method to obtain a BlogData Class. Accepts the Numeric ID of the blog. Returns either a populated BlogData class or null (nothing).

This method can be accessed through the blog.asmx file located in:

~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx

This function is overloaded. See Also:

GetBlogData (FolderData) (see page 1123)

GetBlogData (long, DateTime) (see page 1125)

GetBlogData (long, DateTime, int, int,) (see page 1125)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogData By ID", Description = "")]
public BlogData GetBlogData(long BlogID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogData By ID", Description = "")>
Public Function GetBlogData(BlogID As long) As BlogData
```

Parameters

Parameters	Description
long BlogID	Numeric ID of the blog. This corresponds to the ID of the blog folder.

Returns

Either a populated BlogData class or null (nothing).

Remarks

Blog.GetBlogData Method (long, DateTime)

Overloaded Method to obtain a BlogData Class. Accepts the numeric ID of the blog, as well as the StartDate (as DateTime) for the posts. Returns either a populated BlogData class or null (nothing).

This method can be accessed through the blog.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx`

This function is overloaded. See Also:

GetBlogData (FolderData) (see page 1123)

GetBlogData (Long) (see page 1124)

GetBlogData (Long, DateTime, int, int,) (see page 1125)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogData By ID and StartDate", Description = "")]
public BlogData GetBlogData(long BlogID, DateTime StartDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogData By ID and StartDate", Description = "")>
Public Function GetBlogData(BlogID As long, StartDate As DateTime) As BlogData
```

Parameters

Parameters	Description
long BlogID	Numeric ID of the blog. This corresponds to the ID of the blog folder.
DateTime StartDate	The DateTime value to get posts from. You can obtain historical posts with this property, or leave it blank to begin now.

Returns

Either a populated BlogData class or null (nothing).

Remarks

Blog.GetBlogData Method (long, DateTime, int, int)

Overloaded Method to obtain a BlogData Class. Accepts the numeric ID of the blog, the StartDate (as DateTime) for the posts, the number of posts visible and the language ID. Returns either a populated BlogData class or null (nothing).

This method can be accessed through the blog.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx`

This function is overloaded. See Also:

GetBlogData (FolderData) (see page 1123)

GetBlogData (Long) (see page 1124)

GetBlogData (Long, DateTime) (see page 1125)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogData by ID, StartDate, PostsVisible and LanguageID",
Description = "")]
public BlogData GetBlogData(long BlogID, DateTime StartDate, int Postsvisible, int LanguageID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogData by ID, StartDate, PostsVisible and LanguageID",
Description = "")>
Public Function GetBlogData(BlogID As long, StartDate As DateTime, Postsvisible As Integer,
LanguageID As Integer) As BlogData
```

Parameters

Parameters	Description
long BlogID	Numeric ID of the blog. This corresponds to the ID of the blog folder.
DateTime StartDate	The DateTime value to get posts from. You can obtain historical posts with this property, or leave it blank to begin now.
int Postsvisible	The number of posts to return. Leave as 0 to use the blog setting, or -1 to return none.
int LanguageID	Numeric ID of the language.

Returns

Either a populated BlogData class or null (nothing).

Remarks

Blog.GetBlogDataForPost Method

Returns the blog information for a blog post. This contains information about categories, whether comments are enabled, etc. Accepts the post ID.

This method can be accessed through the blog.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogDataForPost", Description = "")]
public BlogData GetBlogDataForPost(long post_id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogDataForPost", Description = "")>
Public Function GetBlogDataForPost(post_id As long) As BlogData
```

Parameters

Parameters	Description
long post_id	Numeric value of the post ID.

Returns

BlogData class

Blog.GetBlogPostData Method

Gets the Post data for a specific blog post. Accepts the ID of the post. Will return tags, trackback, pingback, and categories for the post.

This method can be accessed through the blog.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogPostData", Description = "")]
public BlogPostData GetBlogPostData(long id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogPostData", Description = "")>
Public Function GetBlogPostData(id As long) As BlogPostData
```

Parameters

Parameters	Description
long id	Numeric value of the post ID.

Returns

BlogPostData Class.

GetBlogRoll Method

This is the overview for the GetBlogRoll method overload.

Description

Returns a populated blog roll class. This blog roll is the list of links for the blog. See the BlogRoll class for the properties and usage of the blog roll.

This is an overloaded method.

- GetBlogRoll (Long) - returns a populated blog roll class by providing the ID of the blog.
- GetBlogRoll (String) - returns a populated blog roll class by providing the CMS400.NET folder path of the blog.

Overload List

	Name	Description
	Blog.GetBlogRoll (long) (see page 1128)	Returns a populated blog roll class. Accepts a blog's numeric ID value. See the BlogRoll (see page 59) class for more information. This method can be accessed through the blog.asmx file located in:
	Blog.GetBlogRoll (string) (see page 1128)	Loads a blog roll for a given path. The path is a string of the blog path, ex: `\Media\Blogs\MyBlog`. This blog roll is the list of links for the blog. See the BlogRoll (see page 59) class for the properties and usage of the blog roll. This method can be accessed through the blog.asmx file located in:

Blog.GetBlogRoll Method (long)

Returns a populated blog roll class. Accepts a blog's numeric ID value. See the BlogRoll (see page 59) class for more information.

This method can be accessed through the blog.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx`

This function is overloaded. See Also:

GetBlogRoll (string) (see page 1128)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogRoll", Description = "")]
public BlogRoll GetBlogRoll(long blog_id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogRoll", Description = "")>
Public Function GetBlogRoll(blog_id As long) As BlogRoll
```

Parameters

Parameters	Description
long blog_id	Numeric ID of the blog. This corresponds to the folder ID of the blog.

Returns

A populated BlogRoll (see page 59) class or a null (nothing).

Remarks

Developers are advised to check for null prior to using this class.

Blog.GetBlogRoll Method (string)

Loads a blog roll for a given path. The path is a string of the blog path, ex: `\Media\Blogs\MyBlog`. This blog roll is the list of links for the blog. See the BlogRoll (see page 59) class for the properties and usage of the blog roll.

This method can be accessed through the blog.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx`

This is an overloaded method. See Also:

GetBlogRoll (int).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogRoll By Path", Description = "")]
public BlogRoll GetBlogRoll(string Path);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogRoll By Path", Description = "")>
Public Function GetBlogRoll(Path As string) As BlogRoll
```

Parameters

Parameters	Description
string Path	The path of the blog. This is a string value that corresponds to the folder path of the blog.

Returns

This method returns a populated BlogRoll (see page 59) class or null (nothing).

Remarks

When using, developers are advised to check for null prior to using.

Blog.GetPost Method

Gets a blog post for a given ID. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData Class.

This method can be accessed through the blog.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPost", Description = "")]
public ContentData GetPost(long post_id, BlogPostData PostInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPost", Description = "")>
Public Function GetPost(post_id As long, PostInfo As BlogPostData) As ContentData
```

Parameters

Parameters	Description
long post_id	The numeric ID for the post.
BlogPostData PostInfo	BlogPostData class representing the blog specific post info. This is also in the metadata.

Returns

ContentData class with a ByRef BlogPostData

Blog.GetPostbyID Method

Gets a blog post for a given ID. Includes trackback, pinback, comments, and other information. Returns BlogPostData Class.

This method can be accessed through the blog.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPostbyID", Description = "")]
public BlogPostData GetPostbyID(long post_id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPostbyID", Description = "")>
Public Function GetPostbyID(post_id As long) As BlogPostData
```

Parameters

Parameters	Description
long post_id	The numeric ID for the post.

Returns

BlogPostData class

Blog.GetUserBlog Method

Gets a blog based on a user's ID.

This method can be accessed through the blog.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserBlog", Description = "retrieves the user's blog folder")]
public long GetUserBlog(long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserBlog", Description = "retrieves the user's blog folder")>
Public Function GetUserBlog(userId As long) As long
```

Parameters

Parameters	Description
long userId	A user's ID.

Blog.UpdateBlogProperties Method

This method updates the blog data.

This method can be accessed through the blog.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Blog.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateBlogProperties", Description = "updates blog data.")]
public bool UpdateBlogProperties(BlogData blog);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateBlogProperties", Description = "updates blog data.")>
Public Function UpdateBlogProperties(blog As BlogData) As bool
```

Parameters

Parameters	Description
BlogData blog	blog data

BusinessRules Class

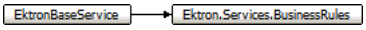
This section describes the Web Service methods used with Ektron's Business Rules. The Business Rules Web service is a wrapper for the Business Rules API (see page 669): `Ektron.Cms.API.BusinessRule` (see page 671). These

methods can be accessed through the BusinessRules.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/BusinessRules.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class BusinessRules : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class BusinessRules
Inherits EktronBaseService
```

File

BusinessRules.cs

BusinessRules Methods

The methods of the BusinessRules class are listed here.

Public Methods

	Name	Description
	GetBusinessRules (see page 1131)	This is the overview for the GetBusinessRules method overload.

GetBusinessRules Method

This is the overview for the GetBusinessRules method overload.

Description

Method to get the results from a business rule. This method is overloaded.

- GetBusinessRules (string, string, string, string, string, string, string, int)
- GetBusinessRules (string, string, string, string, string, string, string, string) (see page 1132)

Overload List

	Name	Description
	BusinessRules.GetBusinessRules (string, string, string, string, string, string, string, long) (see page 1132)	Method to get the results from a business rule by ID. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset. This method can be accessed through the BusinessRules.asmx file located in:
	BusinessRules.GetBusinessRules (string, string, string, string, string, string, string, string) (see page 1132)	Method to get the results from a business rule by identifier. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset. Identifier is a combination of the path/page/uniqueID of the business rules control. This method can be accessed through the BusinessRules.asmx file located in:

BusinessRules.GetBusinessRules Method (string, string, string, string, string, string, string, long)

Method to get the results from a business rule by ID. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset.

This method can be accessed through the BusinessRules.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/BusinessRules.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBusinessRules By Int RuleID", Description = "")]
public string GetBusinessRules(string httpHost, string urlRest, string cookies, string
formValues, string queryValues, string serverValues, string paramValues, long RuleID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBusinessRules By Int RuleID", Description = "")>
Public Function GetBusinessRules(httpHost As string, urlRest As string, cookies As string,
formValues As string, queryValues As string, serverValues As string, paramValues As string,
RuleID As long) As string
```

Parameters

Parameters	Description
string httpHost	The HTTP host the request is made on.
string urlRest	Not used.
string cookies	Cookie Value Collection (see page 95)
string formValues	Form (see page 1247) Value Collection (see page 95)
string queryValues	Querystring Value Collection (see page 95)
string serverValues	Server Variables Collection (see page 95)
string paramValues	Custom Value Collection (see page 95)
long RuleID	Numeric value representing the ID of the Ruleset.

Returns

A string representing the output of the evaluated ruleset.

BusinessRules.GetBusinessRules Method (string, string, string, string, string, string, string, string)

Method to get the results from a business rule by identifier. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset. Identifier is a combination of the path/page/uniqueID of the business rules control.

This method can be accessed through the BusinessRules.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/BusinessRules.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBusinessRules by String RuleID", Description = "")]
public string GetBusinessRules(string httpHost, string urlRest, string cookies, string
formValues, string queryValues, string serverValues, string paramValues, string RuleID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBusinessRules by String RuleID", Description = "")>
Public Function GetBusinessRules(httpHost As string, urlRest As string, cookies As string,
```

```
formValues As string, queryValues As string, serverValues As string, paramValues As string,
RuleID As string) As string
```

Parameters

Parameters	Description
string httpHost	The HTTP host the request is made on.
string urlRest	Not used.
string cookies	Cookie Value Collection (see page 95)
string formValues	Form (see page 1247) Value Collection (see page 95)
string queryValues	Querystring Value Collection (see page 95)
string serverValues	Server Variables Collection (see page 95)
string paramValues	Custom Value Collection (see page 95)
string RuleID	String representing the identifier of the Ruleset. The identifier is a combination of the path/page/uniqueID of the business rules control.

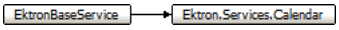
Returns

A string representing the output of the evaluated ruleset.

Calendar Class

Summary description for Calendar

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class Calendar : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class Calendar
Inherits EktronBaseService
```

File

Calendar.cs

Analytics Class

This section describes the Web Service methods used with analytics. The Analytics Web service is a wrapper for the Analytics API (see page 669): `Ektron.Cms.API.Analytics.Analytics` (see page 745). These methods can be accessed through the Analytics.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Analytics/Analytics.asmx`

Web Services (see page 1083) examples involving content are available in the Developer sample site. You can download this site from:

http://www.ektron.com/cms400-web-cms.aspx?id=5278

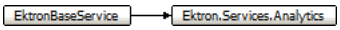
Once the developer site is installed, example code files are located in your site root. For example:

`C:\Inetpub\wwwroot\CMS400Developer\Developer\dotNetAssemblies\WebServicesAPI\AnalyticsAPI`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header. For example:

```
 wscontentApi.AuthenticationHeaderValue = New localhost.AuthenticationHeader()
wscontentApi.AuthenticationHeaderValue.Username = "admin"
wscontentApi.AuthenticationHeaderValue.Password = "admin"
wscontentApi.AuthenticationHeaderValue.Domain = "Domain" '''Only needed if AD/LDAP is being
used
```

Class Hierarchy



C#

```
public class Analytics : EktronBaseService;
```

Visual Basic

```
Public Class Analytics
Inherits EktronBaseService
```

File

Analytics.cs

Description

[WebService(Namespace = "http://ektron.com/webservices/")] [WebServiceBinding(ConformsTo = WsiProfiles.None)]

Analytics Methods

The methods of the Analytics class are listed here.

Public Methods

	Name	Description
	GetAbsoluteUniqueVisitorsTrend (see page 1135)	This is the overview for the GetAbsoluteUniqueVisitorsTrend method overload.
	GetAdVersions (see page 1136)	This is the overview for the GetAdVersions method overload.
	GetAllTrafficSources (see page 1138)	This is the overview for the GetAllTrafficSources method overload.
	GetAveragePageViewsTrend (see page 1139)	This is the overview for the GetAveragePageViewsTrend method overload.
	GetBounceRateTrend (see page 1141)	This is the overview for the GetBounceRateTrend method overload.
	GetBrowsers (see page 1142)	This is the overview for the GetBrowsers method overload.
	GetCampaigns (see page 1143)	This is the overview for the GetCampaigns method overload.
	GetCmsSearchTerms (see page 1145)	a Cms (see page 7) Search (see page 899) terms report This method can be accessed through the Analytics.asmx file located in:
	GetConnectionSpeeds (see page 1145)	This is the overview for the GetConnectionSpeeds method overload.
	GetContentbyTitle (see page 1147)	This is the overview for the GetContentbyTitle method overload.
	GetContentDetail (see page 1148)	This is the overview for the GetContentDetail method overload.
	GetDirectTraffic (see page 1150)	This is the overview for the GetDirectTraffic method overload.
	GetFlashVersions (see page 1151)	This is the overview for the GetFlashVersions method overload.
	GetHostnames (see page 1153)	This is the overview for the GetHostnames method overload.
	GetJavaSupport (see page 1154)	This is the overview for the GetJavaSupport method overload.
	GetKeywords (see page 1155)	This is the overview for the GetKeywords method overload.
	GetLanguages (see page 1157)	This is the overview for the GetLanguages method overload.

	GetLocations (see page 1158)	This is the overview for the GetLocations method overload.
	GetNetworkLocations (see page 1160)	This is the overview for the GetNetworkLocations method overload.
	GetNewVsReturningVisitors (see page 1161)	This is the overview for the GetNewVsReturningVisitors method overload.
	GetOperatingSystems (see page 1163)	This is the overview for the GetOperatingSystems method overload.
	GetPageViewsTrend (see page 1164)	This is the overview for the GetPageViewsTrend method overload.
	GetPlatforms (see page 1165)	This is the overview for the GetPlatforms method overload.
	GetReferringSites (see page 1167)	This is the overview for the GetReferringSites method overload.
	GetScreenColors (see page 1168)	This is the overview for the GetScreenColors method overload.
	GetScreenResolutions (see page 1170)	This is the overview for the GetScreenResolutions method overload.
	GetSearchEngines (see page 1171)	This is the overview for the GetSearchEngines method overload.
	GetTimeOnSiteTrend (see page 1172)	This is the overview for the GetTimeOnSiteTrend method overload.
	GetTopContent (see page 1174)	This is the overview for the GetTopContent method overload.
	GetTopExitPages (see page 1175)	This is the overview for the GetTopExitPages method overload.
	GetTopLandingPages (see page 1177)	This is the overview for the GetTopLandingPages method overload.
	GetUserDefinedValue (see page 1178)	This is the overview for the GetUserDefinedValue method overload.
	GetVisitsTrend (see page 1180)	This is the overview for the GetVisitsTrend method overload.

GetAbsoluteUniqueVisitorsTrend Method

This is the overview for the GetAbsoluteUniqueVisitorsTrend method overload.

Overload List

	Name	Description
	Analytics.GetAbsoluteUniqueVisitorsTrend (DateTime, DateTime, AnalyticsCriteria) (see page 1135)	an absolute unique visitors trend report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetAbsoluteUniqueVisitorsTrend (string, DateTime, DateTime, AnalyticsCriteria) (see page 1136)	an absolute unique visitors trend report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetAbsoluteUniqueVisitorsTrend Method (DateTime, DateTime, AnalyticsCriteria)

an absolute unique visitors trend report by the default provider

This method can be accessed through the Analytics.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAbsoluteUniqueVisitorsTrendReport", Description = "an absolute
unique visitors trend report by the default provider")]
public TrendReportData GetAbsoluteUniqueVisitorsTrend(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAbsoluteUniqueVisitorsTrendReport", Description = "an absolute
unique visitors trend report by the default provider")>
Public Function GetAbsoluteUniqueVisitorsTrend(startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

Analytics.GetAbsoluteUniqueVisitorsTrend Method (string, DateTime, DateTime, AnalyticsCriteria)

an absolute unique visitors trend report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAbsoluteUniqueVisitorsTrendReportByProvider", Description = "an
absolute unique visitors trend report by the specified provider")]
public TrendReportData GetAbsoluteUniqueVisitorsTrend(string provider, DateTime startDate,
DateTime endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAbsoluteUniqueVisitorsTrendReportByProvider", Description = "an
absolute unique visitors trend report by the specified provider")>
Public Function GetAbsoluteUniqueVisitorsTrend(provider As string, startDate As DateTime,
endDate As DateTime, criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

GetAdVersions Method

This is the overview for the GetAdVersions method overload.

Overload List

	Name	Description
	Analytics.GetAdVersions (DateTime, DateTime, AnalyticsCriteria) (see page 1137)	an ad versions report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetAdVersions (string, DateTime, DateTime, AnalyticsCriteria) (see page 1137)	an ad versions report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetAdVersions Method (DateTime, DateTime, AnalyticsCriteria)

an ad versions report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAdVersionsReport", Description = "an ad versions report by the
default provider")]
public AnalyticsReportData GetAdVersions(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAdVersionsReport", Description = "an ad versions report by the
default provider")>
Public Function GetAdVersions(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetAdVersions Method (string, DateTime, DateTime, AnalyticsCriteria)

an ad versions report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAdVersionsReportByProvider", Description = "an ad versions report
by the specified provider")]
```

```
public AnalyticsReportData GetAdVersions(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAdVersionsReportByProvider", Description = "an ad versions report
by the specified provider")>
Public Function GetAdVersions(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetAllTrafficSources Method

This is the overview for the GetAllTrafficSources method overload.

Overload List

	Name	Description
	Analytics.GetAllTrafficSources (DateTime, DateTime, AnalyticsCriteria) (see page 1138)	an all traffic sources report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetAllTrafficSources (string, DateTime, DateTime, AnalyticsCriteria) (see page 1139)	an all traffic sources report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetAllTrafficSources Method (DateTime, DateTime, AnalyticsCriteria)

an all traffic sources report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllTrafficSourcesReport", Description = "an all traffic sources
report by the default provider")]
public AnalyticsReportData GetAllTrafficSources(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllTrafficSourcesReport", Description = "an all traffic sources
report by the default provider")>
Public Function GetAllTrafficSources(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetAllTrafficSources Method (string, DateTime, DateTime, AnalyticsCriteria)

an all traffic sources report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllTrafficSourcesReportByProvider", Description = "an all traffic
sources report by the specified provider")]
public AnalyticsReportData GetAllTrafficSources(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllTrafficSourcesReportByProvider", Description = "an all traffic
sources report by the specified provider")>
Public Function GetAllTrafficSources(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetAveragePageViewsTrend Method

This is the overview for the GetAveragePageViewsTrend method overload.

Overload List

	Name	Description
	Analytics.GetAveragePageViewsTrend (DateTime, DateTime, AnalyticsCriteria) (see page 1140)	an average pageviews for all visitors trend report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetAveragePageViewsTrend (string, DateTime, DateTime, AnalyticsCriteria) (see page 1140)	an average pageviews for all visitors trend report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetAveragePageViewsTrend Method (DateTime, DateTime, AnalyticsCriteria)

an average pageviews for all visitors trend report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAveragePageViewsTrendReport", Description = "an average pageviews
for all visitors trend report by the default provider")]
public TrendReportData GetAveragePageViewsTrend(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAveragePageViewsTrendReport", Description = "an average pageviews
for all visitors trend report by the default provider")>
Public Function GetAveragePageViewsTrend(startDate As DateTime, endDate As DateTime, criteria
As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

Analytics.GetAveragePageViewsTrend Method (string, DateTime, DateTime, AnalyticsCriteria)

an average pageviews for all visitors trend report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAveragePageViewsTrendReportByProvider", Description = "an average
pageviews for all visitors trend report by the specified provider")]
public TrendReportData GetAveragePageViewsTrend(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAveragePageViewsTrendReportByProvider", Description = "an average
pageviews for all visitors trend report by the specified provider")>
Public Function GetAveragePageViewsTrend(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report

DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

GetBounceRateTrend Method

This is the overview for the GetBounceRateTrend method overload.

Overload List

	Name	Description
	Analytics.GetBounceRateTrend (DateTime, DateTime, AnalyticsCriteria) (see page 1141)	a bounce rate for all visitors trend report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetBounceRateTrend (string, DateTime, DateTime, AnalyticsCriteria) (see page 1141)	a bounce rate for all visitors trend report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetBounceRateTrend Method (DateTime, DateTime, AnalyticsCriteria)

a bounce rate for all visitors trend report by the default provider

This method can be accessed through the Analytics.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBounceRateTrendReport", Description = "a bounce rate for all
visitors trend report by the default provider")]
public TrendReportData GetBounceRateTrend(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBounceRateTrendReport", Description = "a bounce rate for all
visitors trend report by the default provider")>
Public Function GetBounceRateTrend(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

Analytics.GetBounceRateTrend Method (string, DateTime, DateTime, AnalyticsCriteria)

a bounce rate for all visitors trend report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBounceRateTrendReportByProvider", Description = "a bounce rate
for all visitors trend report by the specified provider")]
public TrendReportData GetBounceRateTrend(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBounceRateTrendReportByProvider", Description = "a bounce rate
for all visitors trend report by the specified provider")>
Public Function GetBounceRateTrend(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

GetBrowsers Method

This is the overview for the GetBrowsers method overload.

Overload List

	Name	Description
	Analytics.GetBrowsers (DateTime, DateTime, AnalyticsCriteria) (see page 1142)	a browser report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetBrowsers (string, DateTime, DateTime, AnalyticsCriteria) (see page 1143)	a browser report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetBrowsers Method (DateTime, DateTime, AnalyticsCriteria)

a browser report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBrowsersReport", Description = "a browser report by the default
provider")]
```

```
public AnalyticsReportData GetBrowsers(DateTime startDate, DateTime endDate, AnalyticsCriteria
criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBrowsersReport", Description = "a browser report by the default
provider")>
Public Function GetBrowsers(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetBrowsers Method (string, DateTime, DateTime, AnalyticsCriteria)

a browser report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBrowsersReportByProvider", Description = "a browser report by the
specified provider")]
public AnalyticsReportData GetBrowsers(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBrowsersReportByProvider", Description = "a browser report by the
specified provider")>
Public Function GetBrowsers(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetCampaigns Method

This is the overview for the GetCampaigns method overload.

Overload List

	Name	Description
	Analytics.GetCampaigns (DateTime, DateTime, AnalyticsCriteria) (see page 1144)	a campaigns report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetCampaigns (string, DateTime, DateTime, AnalyticsCriteria) (see page 1144)	a campaigns report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetCampaigns Method (DateTime, DateTime, AnalyticsCriteria)

a campaigns report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCampaignsReport", Description = "a campaigns report by the
default provider")]
public AnalyticsReportData GetCampaigns(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCampaignsReport", Description = "a campaigns report by the
default provider")>
Public Function GetCampaigns(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetCampaigns Method (string, DateTime, DateTime, AnalyticsCriteria)

a campaigns report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCampaignsReportByProvider", Description = "a campaigns report by
the specified provider")]
```

```
public AnalyticsReportData GetCampaigns(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCampaignsReportByProvider", Description = "a campaigns report by
the specified provider")>
Public Function GetCampaigns(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetCmsSearchTerms Method

a Cms (see page 7) Search (see page 899) terms report

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCmsSearchTermsReport", Description = "a Cms Search terms report")]
public AnalyticsReportData GetCmsSearchTerms(DateTime startDate, DateTime endDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCmsSearchTermsReport", Description = "a Cms Search terms report")>
Public Function GetCmsSearchTerms(startDate As DateTime, endDate As DateTime) As
AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetConnectionSpeeds Method

This is the overview for the GetConnectionSpeeds method overload.

Overload List

	Name	Description
	Analytics.GetConnectionSpeeds (DateTime, DateTime, AnalyticsCriteria) (see page 1146)	a connection speeds report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetConnectionSpeeds (string, DateTime, DateTime, AnalyticsCriteria) (see page 1146)	a connection speeds report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetConnectionSpeeds Method (DateTime, DateTime, AnalyticsCriteria)

a connection speeds report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetConnectionSpeedsReport", Description = "a connection speeds
report by the default provider")]
public AnalyticsReportData GetConnectionSpeeds(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetConnectionSpeedsReport", Description = "a connection speeds
report by the default provider")>
Public Function GetConnectionSpeeds(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetConnectionSpeeds Method (string, DateTime, DateTime, AnalyticsCriteria)

a connection speeds report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetConnectionSpeedsReportByProvider", Description = "a connection
speeds report by the specified provider")]
public AnalyticsReportData GetConnectionSpeeds(string provider, DateTime startDate, DateTime
```

```
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetConnectionSpeedsReportByProvider", Description = "a connection
speeds report by the specified provider")>
Public Function GetConnectionSpeeds(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetContentbyTitle Method

This is the overview for the GetContentbyTitle method overload.

Overload List

	Name	Description
	Analytics.GetContentbyTitle (DateTime, DateTime, AnalyticsCriteria) (see page 1147)	a content by title report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetContentbyTitle (string, DateTime, DateTime, AnalyticsCriteria) (see page 1148)	a content by title report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetContentbyTitle Method (DateTime, DateTime, AnalyticsCriteria)

a content by title report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentbyTitleReport", Description = "a content by title report
by the default provider")]
public AnalyticsReportData GetContentbyTitle(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentbyTitleReport", Description = "a content by title report
by the default provider")>
Public Function GetContentbyTitle(startDate As DateTime, endDate As DateTime, criteria As
```

```
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetContentbyTitle Method (string, DateTime, DateTime, AnalyticsCriteria)

a content by title report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentbyTitleReportByProvider", Description = "a content by
title report by the specified provider")]
public AnalyticsReportData GetContentbyTitle(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentbyTitleReportByProvider", Description = "a content by
title report by the specified provider")>
Public Function GetContentbyTitle(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetContentDetail Method

This is the overview for the GetContentDetail method overload.

Overload List

	Name	Description
	Analytics.GetContentDetail (DateTime, DateTime, AnalyticsCriteria) (see page 1149)	a content detail report This method can be accessed through the Analytics.asmx file located in:

	Analytics.GetContentDetail (string, DateTime, DateTime, AnalyticsCriteria) (see page 1149)	a content detail report This method can be accessed through the Analytics.asmx file located in:

Analytics.GetContentDetail Method (DateTime, DateTime, AnalyticsCriteria)

a content detail report

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentDetailReport", Description = "a content detail report by
default provider")]
public AnalyticsReportData GetContentDetail(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentDetailReport", Description = "a content detail report by
default provider")>
Public Function GetContentDetail(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetContentDetail Method (string, DateTime, DateTime, AnalyticsCriteria)

a content detail report

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentDetailReportByProvider", Description = "a content detail
report by the specified provider")]
public AnalyticsReportData GetContentDetail(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentDetailReportByProvider", Description = "a content detail
report by the specified provider")>
```

```
Public Function GetContentDetail(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetDirectTraffic Method

This is the overview for the GetDirectTraffic method overload.

Overload List

	Name	Description
	Analytics.GetDirectTraffic (DateTime, DateTime, AnalyticsCriteria) (see page 1150)	a direct traffic report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetDirectTraffic (string, DateTime, DateTime, AnalyticsCriteria) (see page 1151)	a direct traffic report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetDirectTraffic Method (DateTime, DateTime, AnalyticsCriteria)

a direct traffic report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDirectTrafficReport", Description = "a direct traffic report by
the default provider")]
public AnalyticsReportData GetDirectTraffic(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDirectTrafficReport", Description = "a direct traffic report by
the default provider")>
Public Function GetDirectTraffic(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetDirectTraffic Method (string, DateTime, DateTime, AnalyticsCriteria)

a direct traffic report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDirectTrafficReportByProvider", Description = "a direct traffic
report by the specified provider")]
public AnalyticsReportData GetDirectTraffic(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDirectTrafficReportByProvider", Description = "a direct traffic
report by the specified provider")>
Public Function GetDirectTraffic(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetFlashVersions Method

This is the overview for the GetFlashVersions method overload.

Overload List

	Name	Description
	Analytics.GetFlashVersions (DateTime, DateTime, AnalyticsCriteria) (see page 1151)	a flash versions report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetFlashVersions (string, DateTime, DateTime, AnalyticsCriteria) (see page 1152)	a flash versions report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetFlashVersions Method (DateTime, DateTime, AnalyticsCriteria)

a flash versions report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFlashVersionsReport", Description = "a flash versions report by
the default provider")]
public AnalyticsReportData GetFlashVersions(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFlashVersionsReport", Description = "a flash versions report by
the default provider")>
Public Function GetFlashVersions(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetFlashVersions Method (string, DateTime, DateTime, AnalyticsCriteria)

a flash versions report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFlashVersionsReportByProvider", Description = "a flash versions
report by the specified provider")]
public AnalyticsReportData GetFlashVersions(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFlashVersionsReportByProvider", Description = "a flash versions
report by the specified provider")>
Public Function GetFlashVersions(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetHostnames Method

This is the overview for the GetHostnames method overload.

Overload List

	Name	Description
	Analytics.GetHostnames (DateTime, DateTime, AnalyticsCriteria) (see page 1153)	a hostnames report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetHostnames (string, DateTime, DateTime, AnalyticsCriteria) (see page 1153)	a hostnames report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetHostnames Method (DateTime, DateTime, AnalyticsCriteria)

a hostnames report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetHostnamesReport", Description = "a hostnames report by the
default provider")]
public AnalyticsReportData GetHostnames(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetHostnamesReport", Description = "a hostnames report by the
default provider")>
Public Function GetHostnames(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetHostnames Method (string, DateTime, DateTime, AnalyticsCriteria)

a hostnames report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetHostnamesReportByProvider", Description = "a hostnames report by
the specified provider")]
public AnalyticsReportData GetHostnames(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetHostnamesReportByProvider", Description = "a hostnames report by
the specified provider")>
Public Function GetHostnames(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetJavaSupport Method

This is the overview for the GetJavaSupport method overload.

Overload List

	Name	Description
	Analytics.GetJavaSupport (DateTime, DateTime, AnalyticsCriteria) (see page 1154)	a java support report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetJavaSupport (string, DateTime, DateTime, AnalyticsCriteria) (see page 1155)	a java support report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetJavaSupport Method (DateTime, DateTime, AnalyticsCriteria)

a java support report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetJavaSupportReport", Description = "a java support report by the
default provider")]
public AnalyticsReportData GetJavaSupport(DateTime startDate, DateTime endDate,
```

```
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetJavaSupportReport", Description = "a java support report by the
default provider")>
Public Function GetJavaSupport(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetJavaSupport Method (string, DateTime, DateTime, AnalyticsCriteria)

a java support report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetJavaSupportReportByProvider", Description = "a java support
report by the specified provider")]
public AnalyticsReportData GetJavaSupport(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetJavaSupportReportByProvider", Description = "a java support
report by the specified provider")>
Public Function GetJavaSupport(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetKeywords Method

This is the overview for the GetKeywords method overload.

Overload List

	Name	Description
	Analytics.GetKeywords (DateTime, DateTime, AnalyticsCriteria) (see page 1156)	a keywords report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetKeywords (string, DateTime, DateTime, AnalyticsCriteria) (see page 1156)	a keywords report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetKeywords Method (DateTime, DateTime, AnalyticsCriteria)

a keywords report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetKeywordsReport", Description = "a keywords report by the default
provider")]
public AnalyticsReportData GetKeywords(DateTime startDate, DateTime endDate, AnalyticsCriteria
criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetKeywordsReport", Description = "a keywords report by the default
provider")>
Public Function GetKeywords(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetKeywords Method (string, DateTime, DateTime, AnalyticsCriteria)

a keywords report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetKeywordsReportByProvider", Description = "a keywords report by
the specified provider")]
```

```
public AnalyticsReportData GetKeywords(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetKeywordsReportByProvider", Description = "a keywords report by
the specified provider")>
Public Function GetKeywords(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetLanguages Method

This is the overview for the GetLanguages method overload.

Overload List

	Name	Description
	Analytics.GetLanguages (DateTime, DateTime, AnalyticsCriteria) (see page 1157)	a report on different languages used by the visitor by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetLanguages (string, DateTime, DateTime, AnalyticsCriteria) (see page 1158)	a report on different languages used by the visitor by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetLanguages Method (DateTime, DateTime, AnalyticsCriteria)

a report on different languages used by the visitor by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLanguagesReport", Description = "a report on different languages
used by the visitor by the default provider")]
public AnalyticsReportData GetLanguages(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLanguagesReport", Description = "a report on different languages
used by the visitor by the default provider")>
```

```
Public Function GetLanguages(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetLanguages Method (string, DateTime, DateTime, AnalyticsCriteria)

a report on different languages used by the visitor by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLanguagesReportByProvider", Description = "a report on different
languages used by the visitor by the specified provider")]
public AnalyticsReportData GetLanguages(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLanguagesReportByProvider", Description = "a report on different
languages used by the visitor by the specified provider")>
Public Function GetLanguages(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetLocations Method

This is the overview for the GetLocations method overload.

Overload List

	Name	Description
	Analytics.GetLocations (DateTime, DateTime, AnalyticsCriteria) (see page 1159)	an user geographic location report by the default provider This method can be accessed through the Analytics.asmx file located in:

	Analytics.GetLocations (string, DateTime, DateTime, AnalyticsCriteria) (see page 1159)	an user geographic location report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetLocations Method (DateTime, DateTime, AnalyticsCriteria)

an user geographic location report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLocationsReport", Description = "an user geographic location
report by the default provider")]
public AnalyticsReportData GetLocations(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLocationsReport", Description = "an user geographic location
report by the default provider")>
Public Function GetLocations(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetLocations Method (string, DateTime, DateTime, AnalyticsCriteria)

an user geographic location report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLocationsReportByProvider", Description = "an user geographic
location report by the specified provider")]
public AnalyticsReportData GetLocations(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLocationsReportByProvider", Description = "an user geographic
location report by the specified provider")>
```

```
Public Function GetLocations(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetNetworkLocations Method

This is the overview for the GetNetworkLocations method overload.

Overload List

	Name	Description
	Analytics.GetNetworkLocations (DateTime, DateTime, AnalyticsCriteria) (see page 1160)	a network location report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetNetworkLocations (string, DateTime, DateTime, AnalyticsCriteria) (see page 1161)	a network location report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetNetworkLocations Method (DateTime, DateTime, AnalyticsCriteria)

a network location report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetNetworkLocationsReport", Description = "a network location report
by the default provider")]
public AnalyticsReportData GetNetworkLocations(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetNetworkLocationsReport", Description = "a network location report
by the default provider")>
Public Function GetNetworkLocations(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetNetworkLocations Method (string, DateTime, DateTime, AnalyticsCriteria)

a network location report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetNetworkLocationsReportByProvider", Description = "a network
location report by the specified provider")]
public AnalyticsReportData GetNetworkLocations(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetNetworkLocationsReportByProvider", Description = "a network
location report by the specified provider")>
Public Function GetNetworkLocations(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetNewVsReturningVisitors Method

This is the overview for the GetNewVsReturningVisitors method overload.

Overload List

	Name	Description
	Analytics.GetNewVsReturningVisitors (DateTime, DateTime, AnalyticsCriteria) (see page 1161)	a report on new visitor vs. returning visitor by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetNewVsReturningVisitors (string, DateTime, DateTime, AnalyticsCriteria) (see page 1162)	a report on new visitor vs. returning visitor by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetNewVsReturningVisitors Method (DateTime, DateTime, AnalyticsCriteria)

a report on new visitor vs. returning visitor by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetNewVsReturningVisitorsReport", Description = "a report on new
visitor vs. returning visitor by the default provider")]
public AnalyticsReportData GetNewVsReturningVisitors(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetNewVsReturningVisitorsReport", Description = "a report on new
visitor vs. returning visitor by the default provider")>
Public Function GetNewVsReturningVisitors(startDate As DateTime, endDate As DateTime, criteria
As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetNewVsReturningVisitors Method (string, DateTime, DateTime, AnalyticsCriteria)

a report on new visitor vs. returning visitor by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetNewVsReturningVisitorsReportByProvider", Description = "a report
on new visitor vs. returning visitor by the specified provider")]
public AnalyticsReportData GetNewVsReturningVisitors(string provider, DateTime startDate,
DateTime endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetNewVsReturningVisitorsReportByProvider", Description = "a report
on new visitor vs. returning visitor by the specified provider")>
Public Function GetNewVsReturningVisitors(provider As string, startDate As DateTime, endDate
As DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetOperatingSystems Method

This is the overview for the GetOperatingSystems method overload.

Overload List

	Name	Description
	Analytics.GetOperatingSystems (DateTime, DateTime, AnalyticsCriteria) (see page 1163)	an operating systems report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetOperatingSystems (string, DateTime, DateTime, AnalyticsCriteria) (see page 1163)	an operating systems report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetOperatingSystems Method (DateTime, DateTime, AnalyticsCriteria)

an operating systems report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetOperatingSystemsReport", Description = "an operating systems
report by the default provider")]
public AnalyticsReportData GetOperatingSystems(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetOperatingSystemsReport", Description = "an operating systems
report by the default provider")>
Public Function GetOperatingSystems(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetOperatingSystems Method (string, DateTime, DateTime, AnalyticsCriteria)

an operating systems report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
```

```
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetOperatingSystemsReportByProvider", Description = "an operating
systems report by the specified provider")]
public AnalyticsReportData GetOperatingSystems(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetOperatingSystemsReportByProvider", Description = "an operating
systems report by the specified provider")>
Public Function GetOperatingSystems(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetPageViewsTrend Method

This is the overview for the GetPageViewsTrend method overload.

Overload List

	Name	Description
	Analytics.GetPageViewsTrend (DateTime, DateTime, AnalyticsCriteria) (see page 1164)	a pageviews for all visitors trend report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetPageViewsTrend (string, DateTime, DateTime, AnalyticsCriteria) (see page 1165)	a pageviews for all visitors trend report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetPageViewsTrend Method (DateTime, DateTime, AnalyticsCriteria)

a pageviews for all visitors trend report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPageViewsTrendReport", Description = "a pageviews for all
visitors trend report by the default provider")]
public TrendReportData GetPageViewsTrend(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPageViewsTrendReport", Description = "a pageviews for all
```

```
visitors trend report by the default provider")>
Public Function GetPageViewsTrend(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

Analytics.GetPageViewsTrend Method (string, DateTime, DateTime, AnalyticsCriteria)

a pageviews for all visitors trend report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPageViewsTrendReportByProvider", Description = "a pageviews for
all visitors trend report by the specified provider")]
public TrendReportData GetPageViewsTrend(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPageViewsTrendReportByProvider", Description = "a pageviews for
all visitors trend report by the specified provider")>
Public Function GetPageViewsTrend(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

GetPlatforms Method

This is the overview for the GetPlatforms method overload.

Overload List

	Name	Description
	Analytics.GetPlatforms (DateTime, DateTime, AnalyticsCriteria) (see page 1166)	a platforms report by the default provider This method can be accessed through the Analytics.asmx file located in:

	Analytics.GetPlatforms (string, DateTime, DateTime, AnalyticsCriteria) (see page 1166)	a platforms report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetPlatforms Method (DateTime, DateTime, AnalyticsCriteria)

a platforms report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPlatformsReport", Description = "a platforms report by the
default provider")]
public AnalyticsReportData GetPlatforms(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPlatformsReport", Description = "a platforms report by the
default provider")>
Public Function GetPlatforms(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetPlatforms Method (string, DateTime, DateTime, AnalyticsCriteria)

a platforms report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPlatformsReportByProvider", Description = "a platforms report by
the specified provider")]
public AnalyticsReportData GetPlatforms(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPlatformsReportByProvider", Description = "a platforms report by
the specified provider")>
```

```
Public Function GetPlatforms(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetReferringSites Method

This is the overview for the GetReferringSites method overload.

Overload List

	Name	Description
	Analytics.GetReferringSites (DateTime, DateTime, AnalyticsCriteria) (see page 1167)	a referring sites report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetReferringSites (string, DateTime, DateTime, AnalyticsCriteria) (see page 1168)	a referring sites report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetReferringSites Method (DateTime, DateTime, AnalyticsCriteria)

a referring sites report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetReferringSitesReport", Description = "a referring sites report by
the default provider")]
public AnalyticsReportData GetReferringSites(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetReferringSitesReport", Description = "a referring sites report by
the default provider")>
Public Function GetReferringSites(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetReferringSites Method (string, DateTime, DateTime, AnalyticsCriteria)

a referring sites report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetReferringSitesReportByProvider", Description = "a referring sites
report by the specified provider")]
public AnalyticsReportData GetReferringSites(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetReferringSitesReportByProvider", Description = "a referring sites
report by the specified provider")>
Public Function GetReferringSites(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetScreenColors Method

This is the overview for the GetScreenColors method overload.

Overload List

	Name	Description
	Analytics.GetScreenColors (DateTime, DateTime, AnalyticsCriteria) (see page 1168)	a screen colors report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetScreenColors (string, DateTime, DateTime, AnalyticsCriteria) (see page 1169)	a screen colors report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetScreenColors Method (DateTime, DateTime, AnalyticsCriteria)

a screen colors report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetScreenColorsReport", Description = "a screen colors report by the
default provider")]
public AnalyticsReportData GetScreenColors(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetScreenColorsReport", Description = "a screen colors report by the
default provider")>
Public Function GetScreenColors(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetScreenColors Method (string, DateTime, DateTime, AnalyticsCriteria)

a screen colors report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetScreenColorsReportByProvider", Description = "a screen colors
report by the specified provider")]
public AnalyticsReportData GetScreenColors(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetScreenColorsReportByProvider", Description = "a screen colors
report by the specified provider")>
Public Function GetScreenColors(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetScreenResolutions Method

This is the overview for the GetScreenResolutions method overload.

Overload List

	Name	Description
	Analytics.GetScreenResolutions (DateTime, DateTime, AnalyticsCriteria) (see page 1170)	a screen resolutions report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetScreenResolutions (string, DateTime, DateTime, AnalyticsCriteria) (see page 1170)	a screen resolutions report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetScreenResolutions Method (DateTime, DateTime, AnalyticsCriteria)

a screen resolutions report by the default provider

This method can be accessed through the Analytics.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetScreenResolutionsReport", Description = "a screen resolutions
report by the default provider")]
public AnalyticsReportData GetScreenResolutions(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetScreenResolutionsReport", Description = "a screen resolutions
report by the default provider")>
Public Function GetScreenResolutions(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetScreenResolutions Method (string, DateTime, DateTime, AnalyticsCriteria)

a screen resolutions report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetScreenResolutionsReportByProvider", Description = "a screen
resolutions report by the specified provider")]
public AnalyticsReportData GetScreenResolutions(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetScreenResolutionsReportByProvider", Description = "a screen
resolutions report by the specified provider")>
Public Function GetScreenResolutions(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetSearchEngines Method

This is the overview for the GetSearchEngines method overload.

Overload List

	Name	Description
	Analytics.GetSearchEngines (DateTime, DateTime, AnalyticsCriteria) (see page 1171)	"a search engines report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetSearchEngines (string, DateTime, DateTime, AnalyticsCriteria) (see page 1172)	a search engines report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetSearchEngines Method (DateTime, DateTime, AnalyticsCriteria)

"a search engines report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetSearchEnginesReport", Description = "a search engines report by
the default provider")]
public AnalyticsReportData GetSearchEngines(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetSearchEnginesReport", Description = "a search engines report by
the default provider")>
Public Function GetSearchEngines(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetSearchEngines Method (string, DateTime, DateTime, AnalyticsCriteria)

a search engines report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetSearchEnginesReportByProvider", Description = "a search engines
report by the specified provider")]
public AnalyticsReportData GetSearchEngines(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetSearchEnginesReportByProvider", Description = "a search engines
report by the specified provider")>
Public Function GetSearchEngines(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetTimeOnSiteTrend Method

This is the overview for the GetTimeOnSiteTrend method overload.

Overload List

	Name	Description
	Analytics.GetTimeOnSiteTrend (DateTime, DateTime, AnalyticsCriteria) (see page 1173)	a time on site for all visitors trend report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetTimeOnSiteTrend (string, DateTime, DateTime, AnalyticsCriteria) (see page 1173)	a time on site for all visitors trend report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetTimeOnSiteTrend Method (DateTime, DateTime, AnalyticsCriteria)

a time on site for all visitors trend report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTimeOnSiteTrendReport", Description = "a time on site for all
visitors trend report by the default provider")]
public TrendReportData GetTimeOnSiteTrend(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTimeOnSiteTrendReport", Description = "a time on site for all
visitors trend report by the default provider")>
Public Function GetTimeOnSiteTrend(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

Analytics.GetTimeOnSiteTrend Method (string, DateTime, DateTime, AnalyticsCriteria)

a time on site for all visitors trend report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTimeOnSiteTrendReportByProvider", Description = "a time on site
for all visitors trend report by the specified provider")]
public TrendReportData GetTimeOnSiteTrend(string provider, DateTime startDate, DateTime
```

```
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTimeOnSiteTrendReportByProvider", Description = "a time on site
for all visitors trend report by the specified provider")>
Public Function GetTimeOnSiteTrend(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

GetTopContent Method

This is the overview for the GetTopContent method overload.

Overload List

	Name	Description
	Analytics.GetTopContent (DateTime, DateTime, AnalyticsCriteria) (see page 1174)	a top content report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetTopContent (string, DateTime, DateTime, AnalyticsCriteria) (see page 1175)	a top content report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetTopContent Method (DateTime, DateTime, AnalyticsCriteria)

a top content report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTopContentReport", Description = "a top content report by the
default provider")]
public AnalyticsReportData GetTopContent(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTopContentReport", Description = "a top content report by the
default provider")>
Public Function GetTopContent(startDate As DateTime, endDate As DateTime, criteria As
```

```
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetTopContent Method (string, DateTime, DateTime, AnalyticsCriteria)

a top content report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTopContentReportByProvider", Description = "a top content report
by the specified provider")]
public AnalyticsReportData GetTopContent(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTopContentReportByProvider", Description = "a top content report
by the specified provider")>
Public Function GetTopContent(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetTopExitPages Method

This is the overview for the GetTopExitPages method overload.

Overload List

	Name	Description
	Analytics.GetTopExitPages (DateTime, DateTime, AnalyticsCriteria) (see page 1176)	a top exit pages report by the default provider This method can be accessed through the Analytics.asmx file located in:

	Analytics.GetTopExitPages (string, DateTime, DateTime, AnalyticsCriteria) (see page 1176)	a top exit pages report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetTopExitPages Method (DateTime, DateTime, AnalyticsCriteria)

a top exit pages report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTopExitPagesReport", Description = "a top exit pages report by
the default provider")]
public AnalyticsReportData GetTopExitPages(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTopExitPagesReport", Description = "a top exit pages report by
the default provider")>
Public Function GetTopExitPages(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetTopExitPages Method (string, DateTime, DateTime, AnalyticsCriteria)

a top exit pages report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTopExitPagesReportByProvider", Description = "a top exit pages
report by the specified provider")]
public AnalyticsReportData GetTopExitPages(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTopExitPagesReportByProvider", Description = "a top exit pages
report by the specified provider")>
```

```
Public Function GetTopExitPages(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetTopLandingPages Method

This is the overview for the GetTopLandingPages method overload.

Overload List

	Name	Description
	Analytics.GetTopLandingPages (DateTime, DateTime, AnalyticsCriteria) (see page 1177)	a top landing pages report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetTopLandingPages (string, DateTime, DateTime, AnalyticsCriteria) (see page 1178)	a top landing pages report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetTopLandingPages Method (DateTime, DateTime, AnalyticsCriteria)

a top landing pages report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTopLandingPagesReport", Description = "a top landing pages report
by the default provider")]
public AnalyticsReportData GetTopLandingPages(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTopLandingPagesReport", Description = "a top landing pages report
by the default provider")>
Public Function GetTopLandingPages(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetTopLandingPages Method (string, DateTime, DateTime, AnalyticsCriteria)

a top landing pages report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTopLandingPagesReportByProvider", Description = "a top landing
pages report by the specified provider")]
public AnalyticsReportData GetTopLandingPages(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTopLandingPagesReportByProvider", Description = "a top landing
pages report by the specified provider")>
Public Function GetTopLandingPages(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetUserDefinedValue Method

This is the overview for the GetUserDefinedValue method overload.

Overload List

	Name	Description
	Analytics.GetUserDefinedValue (DateTime, DateTime, AnalyticsCriteria) (see page 1178)	an user defined value report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetUserDefinedValue (string, DateTime, DateTime, AnalyticsCriteria) (see page 1179)	an user defined value report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetUserDefinedValue Method (DateTime, DateTime, AnalyticsCriteria)

an user defined value report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserDefinedValueReport", Description = "an user defined value
report by the default provider")]
public AnalyticsReportData GetUserDefinedValue(DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserDefinedValueReport", Description = "an user defined value
report by the default provider")>
Public Function GetUserDefinedValue(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

Analytics.GetUserDefinedValue Method (string, DateTime, DateTime, AnalyticsCriteria)

an user defined value report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserDefinedValueReportByProvider", Description = "an user defined
value report by the specified provider")]
public AnalyticsReportData GetUserDefinedValue(string provider, DateTime startDate, DateTime
endDate, AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserDefinedValueReportByProvider", Description = "an user defined
value report by the specified provider")>
Public Function GetUserDefinedValue(provider As string, startDate As DateTime, endDate As
DateTime, criteria As AnalyticsCriteria) As AnalyticsReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.AnalyticsReportData

GetVisitsTrend Method

This is the overview for the GetVisitsTrend method overload.

Overload List

	Name	Description
	Analytics.GetVisitsTrend (DateTime, DateTime, AnalyticsCriteria) (see page 1180)	a visitors trend report by the default provider This method can be accessed through the Analytics.asmx file located in:
	Analytics.GetVisitsTrend (string, DateTime, DateTime, AnalyticsCriteria) (see page 1180)	a visitors trend report by the specified provider This method can be accessed through the Analytics.asmx file located in:

Analytics.GetVisitsTrend Method (DateTime, DateTime, AnalyticsCriteria)

a visitors trend report by the default provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetVisitsTrendReport", Description = "a visitors trend report by the
default provider")]
public TrendReportData GetVisitsTrend(DateTime startDate, DateTime endDate, AnalyticsCriteria
criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetVisitsTrendReport", Description = "a visitors trend report by the
default provider")>
Public Function GetVisitsTrend(startDate As DateTime, endDate As DateTime, criteria As
AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

Analytics.GetVisitsTrend Method (string, DateTime, DateTime, AnalyticsCriteria)

a visitors trend report by the specified provider

This method can be accessed through the Analytics.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Analytics.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetVisitsTrendReportByProvider", Description = "a visitors trend
report by the specified provider")]
public TrendReportData GetVisitsTrend(string provider, DateTime startDate, DateTime endDate,
AnalyticsCriteria criteria);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetVisitsTrendReportByProvider", Description = "a visitors trend
report by the specified provider")>
Public Function GetVisitsTrend(provider As string, startDate As DateTime, endDate As DateTime,
criteria As AnalyticsCriteria) As TrendReportData
```

Parameters

Parameters	Description
string provider	the provider to carry out the report
DateTime startDate	start date of the report
DateTime endDate	end date of the report
AnalyticsCriteria criteria	the conditions for the report query

Returns

Ektron.Cms.Analytics.Reporting.TrendReportData

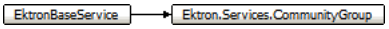
CommunityGroup Class

This section describes the Web Service methods used with Ektron's Community (see page 750) Groups. The Community (see page 750) Groups Web service is a wrapper for the Community (see page 750) Groups API (see page 669): `Ektron.Cms.API.Community.CommunityGroup` (see page 750). These methods can be accessed through the CommunityGroup.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
public class CommunityGroup : EktronBaseService;
```

Visual Basic

```
Public Class CommunityGroup
Inherits EktronBaseService
```

File

CommunityGroup.cs

CommunityGroup Methods

The methods of the CommunityGroup class are listed here.

Public Methods

	Name	Description
	AddCommunityGroup (see page 1182)	Saves a new community group. This method can be accessed through the CommunityGroup.asmx file located in:
	AddUserToCommunityGroup (see page 1183)	This is the overview for the AddUserToCommunityGroup method overload.
	ApprovePendingGroupUser (see page 1184)	Approves the user's pending group request. This method can be accessed through the CommunityGroup.asmx file located in:
	CancelJoinRequestForCommunityGroup (see page 1185)	Cancels a community group join request sent by the user. This method can be accessed through the CommunityGroup.asmx file located in:
	DeleteCommunityGroupByID (see page 1185)	Deletes an existing community group. This method can be accessed through the CommunityGroup.asmx file located in:
	DeletePendingGroupUser (see page 1186)	Delete or reject a users pending group request. This method can be accessed through the CommunityGroup.asmx file located in:
	GetAllCommunityGroups (see page 1186)	Returns an array of Community (see page 750) Groups based upon supplied criteria. This method can be accessed through the CommunityGroup.asmx file located in:
	GetAllCommunityGroupsInDirectory (see page 1186)	Retrieves a list of community groups in a Directory (see page 196). This method can be accessed through the CommunityGroup.asmx file located in:
	GetAllUnassignedCommunityGroupUsers (see page 1187)	Returns a list of users who are not currently in a community group. This method can be accessed through the CommunityGroup.asmx file located in:
	GetCommunityGroupBlogIds (see page 1187)	Gets a list of blog IDs that belong to the community group. This method can be accessed through the CommunityGroup.asmx file located in:
	GetCommunityGroupByID (see page 1188)	Retrieves a CommunityGroupData object by ID. This method can be accessed through the CommunityGroup.asmx file located in:
	GetCommunityGroupDiscussionBoardIds (see page 1188)	Gets a list of Discussion Board IDs that belong to the community group. This method can be accessed through the CommunityGroup.asmx file located in:
	GetCommunityGroups (see page 1189)	Gets a list of community groups a user belongs to. This method can be accessed through the CommunityGroup.asmx file located in:
	GetCommunityGroupUserIDs (see page 1189)	Gets a list of user IDs that belong to the community group. This method can be accessed through the CommunityGroup.asmx file located in:
	GetCommunityGroupUsers (see page 1190)	This is the overview for the GetCommunityGroupUsers method overload.
	GetFolderIdForCommunityGroup (see page 1191)	Gets the folder ID used by the Community (see page 750) Group in the CMS content tree. This method can be accessed through the CommunityGroup.asmx file located in:
	GetGroupMemberStatus (see page 1192)	Gets the Member status of a user for a particular group. This method can be accessed through the CommunityGroup.asmx file located in:
	GetInvitedCommunityGroups (see page 1192)	Returns a list of Community (see page 750) Groups that a user has been invited to join. This method can be accessed through the CommunityGroup.asmx file located in:
	GetMostPopularCommunityGroups (see page 1193)	This is the overview for the GetMostPopularCommunityGroups method overload.
	GetPendingCommunityGroups (see page 1194)	Gets a list of community groups for which the user's membership is pending the Group administrator's approval. This method can be accessed through the CommunityGroup.asmx file located in:
	GetPendingCommunityGroupUsers (see page 1195)	Returns a list of users whose community group membership is pending approval. This method can be accessed through the CommunityGroup.asmx file located in:
	InviteUsers (see page 1195)	Invites the users to the community group. This method can be accessed through the CommunityGroup.asmx file located in:
	IsCommunityGroupAdmin (see page 1196)	Returns true if the user is an administrator of the group. This method can be accessed through the CommunityGroup.asmx file located in:
	IsUserInCommunityGroup (see page 1196)	Returns true if the user is a member of the community group. This method can be accessed through the CommunityGroup.asmx file located in:
	RemoveUserFromCommunityGroup (see page 1197)	Removes a user from a community group. This method can be accessed through the CommunityGroup.asmx file located in:
	UpdateCommunityGroup (see page 1197)	Updates an existing community group. This method can be accessed through the CommunityGroup.asmx file located in:

CommunityGroup.AddCommunityGroup Method

Saves a new community group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddCommunityGroup", Description = "Saves a new community group.")]
public long AddCommunityGroup(CommunityGroupData groupData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddCommunityGroup", Description = "Saves a new community group.")>
Public Function AddCommunityGroup(groupData As CommunityGroupData) As long
```

Parameters

Parameters	Description
CommunityGroupData groupData	CommunityGroupData class containing information about new community group.

Returns

ID of the new community group.

AddUserToCommunityGroup Method

This is the overview for the AddUserToCommunityGroup method overload.

Description

This method is overloaded.

- AddUserToCommunityGroup (Long, Long) - adds a user to a community group by providing the group's ID and the user's ID.
- AddUserToCommunityGroup (Long, Long, Bool) - adds a user to a community group by providing the group's ID and the user's ID. This method allows you to override the community group's enrollment moderation.

Overload List

	Name	Description
	CommunityGroup.AddUserToCommunityGroup (long, long) (see page 1183)	Adds a user to a community group by providing the group's ID and the user's ID. This method can be accessed through the CommunityGroup.asmx file located in:
	CommunityGroup.AddUserToCommunityGroup (long, long, bool) (see page 1184)	Adds a user to a community group. This method allows you to bypass community group enrollment moderation. This method can be accessed through the CommunityGroup.asmx file located in:

CommunityGroup.AddUserToCommunityGroup Method (long, long)

Adds a user to a community group by providing the group's ID and the user's ID.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "AddUserToCommunityGroup", Description = "Adds a user to a community
group.")]
public void AddUserToCommunityGroup(long groupId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddUserToCommunityGroup", Description = "Adds a user to a community
group.")>
Public Function AddUserToCommunityGroup(groupId As long, userId As long) As void
```

Parameters

Parameters	Description
long groupId	ID of the community group.
long userId	ID of the user to be added.

CommunityGroup.AddUserToCommunityGroup Method (long, long, bool)

Adds a user to a community group. This method allows you to bypass community group enrollment moderation.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddUserToCommunityGroup with override", Description = "Adds a user
to a community group.")]
public void AddUserToCommunityGroup(long groupId, long userId, bool overridePending);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddUserToCommunityGroup with override", Description = "Adds a user
to a community group.")>
Public Function AddUserToCommunityGroup(groupId As long, userId As long, overridePending As
bool) As void
```

Parameters

Parameters	Description
long groupId	ID of the community group.
long userId	ID of the user to be added.
bool overridePending	Set to true if you wish to bypass community group enrollment moderation.

CommunityGroup.ApprovePendingGroupUser Method

Approves the user's pending group request.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ApprovePendingGroupUser", Description = "Approves the user's pending
group request.")]
public void ApprovePendingGroupUser(long userId, long groupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ApprovePendingGroupUser", Description = "Approves the user's pending
group request.")>
Public Function ApprovePendingGroupUser(userId As long, groupId As long) As void
```

Parameters

Parameters	Description
long userId	ID of the user to approve.
long groupId	ID of the community group to approve user for.

CommunityGroup.CancelJoinRequestForCommunityGroup Method

Cancels a community group join request sent by the user.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CancelJoinRequestForCommunityGroup", Description = "Cancels a group
join request sent by the user.")]
public void CancelJoinRequestForCommunityGroup(long groupId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CancelJoinRequestForCommunityGroup", Description = "Cancels a group
join request sent by the user.")>
Public Function CancelJoinRequestForCommunityGroup(groupId As long, userId As long) As void
```

Parameters

Parameters	Description
long groupId	ID of the community group the join request was sent to.
long userId	ID of the user whom initially sent join request.

CommunityGroup.DeleteCommunityGroupByID Method

Deletes an existing community group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteCommunityGroupByID", Description = "Deletes an existing
community group.")]
public void DeleteCommunityGroupByID(long groupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteCommunityGroupByID", Description = "Deletes an existing
community group.")>
Public Function DeleteCommunityGroupByID(groupId As long) As void
```

Parameters

Parameters	Description
long groupId	ID of the group to delete.

CommunityGroup.DeletePendingGroupUser Method

Delete or reject a users pending group request.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeletePendingGroupUser", Description = "Delete or reject a users
pending group request.")]
public void DeletePendingGroupUser(long userId, long groupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeletePendingGroupUser", Description = "Delete or reject a users
pending group request.")>
Public Function DeletePendingGroupUser(userId As long, groupId As long) As void
```

Parameters

Parameters	Description
long userId	ID of the user for which to delete a request.
long groupId	ID of the community group for which to delete a request.

CommunityGroup.GetAllCommunityGroups Method

Returns an array of Community (see page 750) Groups based upon supplied criteria.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllCommunityGroups", Description = "Returns an array of Community
Groups based upon supplied criteria.")]
public CommunityGroupData[] GetAllCommunityGroups(CommunityGroupRequest request);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllCommunityGroups", Description = "Returns an array of Community
Groups based upon supplied criteria.")>
Public Function GetAllCommunityGroups(request As CommunityGroupRequest) As CommunityGroupData()
```

Parameters

Parameters	Description
CommunityGroupRequest request	Request criteria defining the groups to retrieve.

Returns

An array of CommunityGroupData objects.

CommunityGroup.GetAllCommunityGroupsInDirectory Method

Retrieves a list of community groups in a Directory (see page 196).

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllCommunityGroupsInDirectory", Description = "Retrieves a list
of community groups in a Directory.")]
public CommunityGroupData[] GetAllCommunityGroupsInDirectory(ref DirectoryRequest request);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllCommunityGroupsInDirectory", Description = "Retrieves a list
of community groups in a Directory.")>
Public Function GetAllCommunityGroupsInDirectory(ByRef request As DirectoryRequest) As
CommunityGroupData()
```

Parameters

Parameters	Description
ref DirectoryRequest request	Request criteria for group retrieval.

Returns

An array of CommunityGroupData objects.

CommunityGroup.GetAllUnassignedCommunityGroupUsers Method

Returns a list of users who are not currently in a community group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUnassignedCommunityGroupUsers", Description = "Returns a list
of users who are not currently in a community group.")]
public DirectoryUserData[] GetAllUnassignedCommunityGroupUsers(ref CommunityGroupObjectRequest
request);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUnassignedCommunityGroupUsers", Description = "Returns a list
of users who are not currently in a community group.")>
Public Function GetAllUnassignedCommunityGroupUsers(ByRef request As
CommunityGroupObjectRequest) As DirectoryUserData()
```

Parameters

Parameters	Description
ref CommunityGroupObjectRequest request	Request criteria for user list.

Returns

An array of DirectoryUserData objects.

CommunityGroup.GetCommunityGroupBlogIds Method

Gets a list of blog IDs that belong to the community group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroupBlogIds", Description = "Gets a list of Blog IDs
that belong to the community group.")]
public long[] GetCommunityGroupBlogIds(long communityGroupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroupBlogIds", Description = "Gets a list of Blog IDs
that belong to the community group.")>
Public Function GetCommunityGroupBlogIds(communityGroupId As long) As long()
```

Parameters

Parameters	Description
long communityGroupId	ID of community group.

Returns

An array of blog IDs (Long).

CommunityGroup.GetCommunityGroupByID Method

Retrieves a CommunityGroupData object by ID.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroupByID", Description = "Retrieves a
CommunityGroupData object by ID.")]
public CommunityGroupData GetCommunityGroupByID(long groupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroupByID", Description = "Retrieves a
CommunityGroupData object by ID.")>
Public Function GetCommunityGroupByID(groupId As long) As CommunityGroupData
```

Parameters

Parameters	Description
long groupId	ID of community group to retrieve.

Returns

CommunityGroupData if groupId is valid, else null.

CommunityGroup.GetCommunityGroupDiscussionBoardIds Method

Gets a list of Discussion Board IDs that belong to the community group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
```

```
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroupDiscussionBoardIds", Description = "Gets a list of
Discussion Board IDs that belong to the community group.")]
public long[] GetCommunityGroupDiscussionBoardIds(long communityGroupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroupDiscussionBoardIds", Description = "Gets a list of
Discussion Board IDs that belong to the community group.")>
Public Function GetCommunityGroupDiscussionBoardIds(communityGroupId As long) As long()
```

Parameters

Parameters	Description
long communityGroupId	ID of community group.

Returns

An array of Discussion Board IDs (Long).

CommunityGroup.GetCommunityGroups Method

Gets a list of community groups a user belongs to.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroups", Description = "Gets a list of community groups
a user belongs to.")]
public CommunityGroupData[] GetCommunityGroups(long userId, PagingInfo pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroups", Description = "Gets a list of community groups
a user belongs to.")>
Public Function GetCommunityGroups(userId As long, pagingInfo As PagingInfo) As
CommunityGroupData()
```

Parameters

Parameters	Description
long userId	ID of a user to retrieve community groups for.
PagingInfo pagingInfo	Paging information for list retrieval.

Returns

An array of CommunityGroupData objects.

CommunityGroup.GetCommunityGroupUserIDs Method

Gets a list of user IDs that belong to the community group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "GetCommunityGroupUserIDs", Description = "Gets a list of User IDs
that belong to the community group.")]
public long[] GetCommunityGroupUserIDs(long communityGroupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroupUserIDs", Description = "Gets a list of User IDs
that belong to the community group.")>
Public Function GetCommunityGroupUserIDs(communityGroupId As long) As long()
```

Parameters

Parameters	Description
long communityGroupId	ID of community group.

Returns

An array of discussion board IDs (Long).

GetCommunityGroupUsers Method

This is the overview for the GetCommunityGroupUsers method overload.

Description

This method is overloaded.

- GetCommunityGroupUsers (Long) - retrieves a list of users in the community group.
- GetCommunityGroupUsers (Long, PagingInfo) - retrieves a list of users in the community group and allows you to page the results.

Overload List

	Name	Description
	CommunityGroup.GetCommunityGroupUsers (long) (see page 1190)	Retrieves a list of users in the community group. This method can be accessed through the CommunityGroup.asmx file located in:
	CommunityGroup.GetCommunityGroupUsers (long, PagingInfo) (see page 1191)	Returns a list of users in a community group. This method can be accessed through the CommunityGroup.asmx file located in:

CommunityGroup.GetCommunityGroupUsers Method (long)

Retrieves a list of users in the community group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroupUsers", Description = "Retrieves a list of users in
the community group.")]
public DirectoryUserData[] GetCommunityGroupUsers(long communityGroupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "GetCommunityGroupUsers", Description = "Retrieves a list of users in
the community group.")>
Public Function GetCommunityGroupUsers(communityGroupId As long) As DirectoryUserData()
```

Parameters

Parameters	Description
long communityGroupId	ID of the community group whose users are being retrieved.

Returns

An array of DirectoryUserData.

CommunityGroup.GetCommunityGroupUsers Method (long, PagingInfo)

Returns a list of users in a community group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroupUsers with paging info", Description = "Returns a
list of users in a community group.")]
public DirectoryUserData[] GetCommunityGroupUsers(long communityGroupId, PagingInfo
pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroupUsers with paging info", Description = "Returns a
list of users in a community group.")>
Public Function GetCommunityGroupUsers(communityGroupId As long, pagingInfo As PagingInfo) As
DirectoryUserData()
```

Parameters

Parameters	Description
long communityGroupId	ID of community group.
PagingInfo pagingInfo	Paging information for the user list.

Returns

An array of DirectoryUserData objects.

CommunityGroup.GetFolderIdForCommunityGroup Method

Gets the folder ID used by the Community (see page 750) Group in the CMS content tree.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolderIdForCommunityGroup", Description = "Gets the folder ID
used by the Community Group in the CMS content tree.")]
public long GetFolderIdForCommunityGroup(long CommunityGroupID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "GetFolderIdForCommunityGroup", Description = "Gets the folder ID
used by the Community Group in the CMS content tree.")>
Public Function GetFolderIdForCommunityGroup(CommunityGroupID As long) As long
```

Parameters

Parameters	Description
long CommunityGroupID	The ID of the community group.

Returns

Retuns a Folder (see page 1236) ID (Long).

CommunityGroup.GetGroupMemberStatus Method

Gets the Member status of a user for a particular group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetGroupMemberStatus", Description = "Gets the Member status of a
user for a particular group.")]
public EkEnumeration.GroupMemberStatus GetGroupMemberStatus(long communityGroupId, long
userID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetGroupMemberStatus", Description = "Gets the Member status of a
user for a particular group.")>
Public Function GetGroupMemberStatus(communityGroupId As long, userID As long) As
EkEnumeration.GroupMemberStatus
```

Parameters

Parameters	Description
long communityGroupId	ID of community group to get user status for.
long userID	ID of the user.

Returns

GroupMemberStatus enumeration

CommunityGroup.GetInvitedCommunityGroups Method

Returns a list of Community (see page 750) Groups that a user has been invited to join.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetInvitedCommunityGroups", Description = "Returns a list of
Community Groups that a user has been invited to.")]
public CommunityGroupData[] GetInvitedCommunityGroups(long userId, PagingInfo pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "GetInvitedCommunityGroups", Description = "Returns a list of
Community Groups that a user has been invited to.")>
Public Function GetInvitedCommunityGroups(userId As long, pagingInfo As PagingInfo) As
CommunityGroupData()
```

Parameters

Parameters	Description
long userId	ID of the user who was invited to join the community group.
PagingInfo pagingInfo	Object to supply paging parameters.

Returns

An array of CommunityGroupData.

GetMostPopularCommunityGroups Method

This is the overview for the GetMostPopularCommunityGroups method overload.

Description

This method is overloaded.

- GetMostPopularCommunityGroups (PagingInfo) (see page 1193) - retrieves all community groups ranked by popularity with the ability to have paging. The popularity of a group is determined by how many members belong to the group.
- GetMostPopularCommunityGroups (int) (see page 1194) - retrieves a list of the most popular community groups up to the number entered in the maxResults property. The popularity of a group is determined by how many members belong to the group.

Overload List

	Name	Description
	CommunityGroup.GetMostPopularCommunityGroups (int) (see page 1194)	Retrieves a list of the most popular community groups up to the number entered in the maxResults property. The popularity of a community group is determined by how many members belong to the group. This method can be accessed through the CommunityGroup.asmx file located in:
	CommunityGroup.GetMostPopularCommunityGroups (PagingInfo) (see page 1193)	Retrieves all community groups ranked by popularity with the ability to have paging. The popularity of a group is determined by how many members belong to the group. This method can be accessed through the CommunityGroup.asmx file located in:

CommunityGroup.GetMostPopularCommunityGroups Method (PagingInfo)

Retrieves all community groups ranked by popularity with the ability to have paging. The popularity of a group is determined by how many members belong to the group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMostPopularCommunityGroups with paging info", Description =
"Retrieves the most popular community groups.")]
public CommunityGroupData[] GetMostPopularCommunityGroups(PagingInfo pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMostPopularCommunityGroups with paging info", Description =
"Retrieves the most popular community groups.")>
Public Function GetMostPopularCommunityGroups(pagingInfo As PagingInfo) As CommunityGroupData()
```

Parameters

Parameters	Description
PagingInfo pagingInfo	Paging information for paging through large list of groups.

Returns

An array of CommunityGroupData objects.

CommunityGroup.GetMostPopularCommunityGroups Method (int)

Retrieves a list of the most popular community groups up to the number entered in the maxResults property. The popularity of a community group is determined by how many members belong to the group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMostPopularCommunityGroups", Description = "Retrieves the most
popular community groups.")]
public CommunityGroupData[] GetMostPopularCommunityGroups(int maxResults);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMostPopularCommunityGroups", Description = "Retrieves the most
popular community groups.")>
Public Function GetMostPopularCommunityGroups(maxResults As Integer) As CommunityGroupData()
```

Parameters

Parameters	Description
int maxResults	The number of groups to retrieve. For example, if 10 is supplied, the top 10 most popular groups will be returned.

Returns

An array of CommunityGroupData objects.

CommunityGroup.GetPendingCommunityGroups Method

Gets a list of community groups for which the user's membership is pending the Group administrator's approval.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPendingCommunityGroups", Description = "Gets a list of community
groups for which the user's membership is pending the Group administrator's approval.")]
public CommunityGroupData[] GetPendingCommunityGroups(long userId, PagingInfo pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPendingCommunityGroups", Description = "Gets a list of community
groups for which the user's membership is pending the Group administrator's approval.")>
Public Function GetPendingCommunityGroups(userId As long, pagingInfo As PagingInfo) As
CommunityGroupData()
```

Parameters

Parameters	Description
long userId	ID of user to retrieve pending groups for.
PagingInfo pagingInfo	Paging information for retrieval.

CommunityGroup.GetPendingCommunityGroupUsers Method

Returns a list of users whose community group membership is pending approval.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPendingCommunityGroupUsers", Description = "Returns a list of
users who's community group membership is pending approval.")]
public DirectoryUserData[] GetPendingCommunityGroupUsers(long communityGroupID, PagingInfo
pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPendingCommunityGroupUsers", Description = "Returns a list of
users who's community group membership is pending approval.")>
Public Function GetPendingCommunityGroupUsers(communityGroupID As long, pagingInfo As
PagingInfo) As DirectoryUserData()
```

Parameters

Parameters	Description
PagingInfo pagingInfo	Paging information for the user list.
communityGroupId	ID of community group.

Returns

An array of DirectoryUserData objects.

CommunityGroup.InviteUsers Method

Invites the users to the community group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "InviteUsers", Description = "Invites the users to the community
group.")]
public Collection<Invitation> InviteUsers(InvitationSendRequestData invitationData, long
groupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "InviteUsers", Description = "Invites the users to the community
group.")>
Public Function InviteUsers(invitationData As InvitationSendRequestData, groupId As long) As
Collection<Invitation>
```

Parameters

Parameters	Description
InvitationSendRequestData invitationData	InvitationSendRequestData containing information about the GroupInvitation to send.
long groupId	ID of the community group to invite people to join.

Returns

A collection of invitations that were sent out.

CommunityGroup.IsCommunityGroupAdmin Method

Returns true if the user is an administrator of the group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsCommunityGroupAdmin", Description = "Returns true if the user is
an admin of the group.")]
public bool IsCommunityGroupAdmin(long userId, long groupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsCommunityGroupAdmin", Description = "Returns true if the user is
an admin of the group.")>
Public Function IsCommunityGroupAdmin(userId As long, groupId As long) As bool
```

Parameters

Parameters	Description
long userId	ID of the user to verify.
long groupId	ID of the group to check.

CommunityGroup.IsUserInCommunityGroup Method

Returns true if the user is a member of the community group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsUserInCommunityGroup", Description = "Returns true if the  user is
a member of the community group.")]
public bool IsUserInCommunityGroup(long communityGroupId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "IsUserInCommunityGroup", Description = "Returns true if the  user is
a member of the community group.")>
Public Function IsUserInCommunityGroup(communityGroupId As long, userId As long) As bool
```

Parameters

Parameters	Description
long communityGroupId	ID of community group to check membership.
long userId	User (see page 1369) ID to check for.

Returns

True if the user is a member of the community group.

CommunityGroup.RemoveUserFromCommunityGroup Method

Removes a user from a community group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RemoveUserFromCommunityGroup", Description = "Removes a user from a
community group.")]
public void RemoveUserFromCommunityGroup(long groupId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RemoveUserFromCommunityGroup", Description = "Removes a user from a
community group.")>
Public Function RemoveUserFromCommunityGroup(groupId As long, userId As long) As void
```

Parameters

Parameters	Description
long groupId	ID of the community group.
long userId	ID of the user to remove.

CommunityGroup.UpdateCommunityGroup Method

Updates an existing community group.

This method can be accessed through the CommunityGroup.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/CommunityGroup.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateCommunityGroup", Description = "Updates an existing community
group.")]
public void UpdateCommunityGroup(CommunityGroupData groupData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateCommunityGroup", Description = "Updates an existing community
group.")>
Public Function UpdateCommunityGroup(groupData As CommunityGroupData) As void
```

Parameters

Parameters	Description
CommunityGroupData groupData	CommunityGroupData class containing information to update for the group.

Content Class

This section describes the Web Service methods used with content. The Content Web service is a wrapper for the Content API (see page 669): `Ektron.Cms.API.Content.Content` (see page 840). These methods can be accessed through the Content.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Content.asmx`

Web Services (see page 1083) examples involving content are available in the Developer sample site. You can download this site from:

http://www.ektron.com/cms400-web-cms.aspx?id=5278

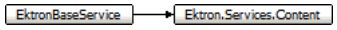
Once the developer site is installed, example code files are located in your site root. For example:

`C:\Inetpub\wwwroot\CMS400Developer\Developer\dotNetAssemblies\WebServicesAPI\ContentAPI\Content`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header. For example:

```
 wscontentApi.AuthenticationHeaderValue = New localhost.AuthenticationHeader()
wscontentApi.AuthenticationHeaderValue.Username = "admin"
wscontentApi.AuthenticationHeaderValue.Password = "admin"
wscontentApi.AuthenticationHeaderValue.Domain = "Domain" '''Only needed if AD/LDAP is being
used
```

Class Hierarchy



C#

```
public class Content : EktronBaseService;
```

Visual Basic

```
Public Class Content
Inherits EktronBaseService
```

File

Content.cs

Description

[WebService(Namespace = "http://ektron.com/webservices/")] [WebServiceBinding(ConformsTo = WsiProfiles.None)]

Content Methods

The methods of the Content class are listed here.

Public Methods

	Name	Description
	AddContent (see page 1199)	This is the overview for the AddContent method overload.
	CheckOutContent (see page 1203)	This will check the content out to the current user if the content is currently checked in and the user has privileges to check it out. This method can be accessed through the Content.asmx file located in:
	CopyContentToFolder (see page 1203)	Copy the content into required folder. This method can be accessed through the Content.asmx file located in:

	DeleteContentItem (see page 1204)	Delete the content by the content's ID. This method can be accessed through the Content.asmx file located in:
	GetAddViewLanguage (see page 1204)	Loads all the languages from the system for a given content ID and populates the Add/View drop down box with flag set on each item. This method can be accessed through the Content.asmx file located in:
	GetAllComments (see page 1205)	Loads all of the comments. This method can be accessed through the Content.asmx file located in:
	GetAllTemplates (see page 1205)	Loads all of the templates. This method can be accessed through the Content.asmx file located in:
	GetAllUnassignedItemApprovals (see page 1206)	Returns all of the unassigned user/groups for a given content ID. This method can be accessed through the Content.asmx file located in:
	GetChildContent (see page 1206)	This is the overview for the GetChildContent method overload.
	GetContent (see page 1208)	Get Content (see page 1198) By ID. This method can be accessed through the Content.asmx file located in:
	GetContentByHistory (see page 1208)	Retrieves the corresponding content by the history ID. This method can be accessed through the Content.asmx file located in:
	GetContentForEditing (see page 1209)	Loads the content details with checkout mode. This method can be accessed through the Content.asmx file located in:
	GetContentState (see page 1209)	Loads the content's details. This method can be accessed through the Content.asmx file located in:
	GetContentStatus (see page 1210)	This will return a single letter string which is the status of the piece of content. The letters returned are the same letters used in the Workarea view. This method can be accessed through the Content.asmx file located in:
	GetDomain (see page 1210)	Gets web domain for a piece of content. This method can be accessed through the Content.asmx file located in:
	MoveContentToFolder (see page 1210)	Moves content to a folder. This method can be accessed through the Content.asmx file located in:
	PublishContent (see page 1211)	Function takes in information about the content to be checked in and published. This goes through the normal approval chain. The content must be in a checked out state to the current logged in user for this to succeed. This method can be accessed through the Content.asmx file located in:
	RenameContent (see page 1211)	Renames the content. This method can be accessed through the Content.asmx file located in:
	SaveContent (see page 1212)	Updates the checked out content block. Only following properties can be updated: • contentData.Title • contentData.Teaser • contentData.Comment • contentData.GoLive • contentData.Html • contentData.EndDate • contentData.EndDateAction • contentData.ManualAlias This method can be accessed through the Content.asmx file located in:
	SubmitForDelete (see page 1213)	Delete the content by the content's ID and folder ID. This method can be accessed through the Content.asmx file located in:
	UndoCheckout (see page 1213)	This undoes the checkout for a piece of content which is checked out to the current user. This method can be accessed through the Content.asmx file located in:
	UpdateContentMetaData (see page 1213)	Updates a specific metadata item attached to a specified piece of content. This method can be accessed through the Content.asmx file located in:

AddContent Method

This is the overview for the AddContent method overload.

Description

Adds Content to the CMS. This method is overloaded.

Overload List

	Name	Description
	Content.AddContent (string, string, string, string, string, string, long, object, object, string) (see page 1200)	Adds Content (see page 1198) to the CMS This method can be accessed through the Content.asmx file located in:
	Content.AddContent (string, string, string, string, string, string, long, object, object, string, long) (see page 1201)	Adds Content (see page 1198) to the CMS This method can be accessed through the Content.asmx file located in:
	Content.AddContent (string, string, string, string, string, string, long, object, object, string, long, long) (see page 1202)	Adds Content (see page 1198) to the CMS This method can be accessed through the Content.asmx file located in:
	Content.AddContent (string, string, string, string, string, string, string, string, string, long, object, object, string) (see page 1202)	Adds Content (see page 1198) to the CMS This method can be accessed through the Content.asmx file located in:

Content.AddContent Method (string, string, string, string, string, string, long, object, object, string)

Adds Content (see page 1198) to the CMS

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContent", Description = "Adds a piece of content.")]
public long AddContent(string ContentTitle, string ContentComment, string ContentHtml, string
SearchText, string SummaryHtml, string ContentLanguage, long FolderId, object GoLive, object
EndDate, string MetaInfoXml);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContent", Description = "Adds a piece of content.")>
Public Function AddContent(ContentTitle As string, ContentComment As string, ContentHtml As
string, SearchText As string, SummaryHtml As string, ContentLanguage As string, FolderId As
long, GoLive As object, EndDate As object, MetaInfoXml As string) As long
```

Parameters

Parameters	Description
string ContentTitle	Title of content to add
string ContentComment	Content (see page 1198) comments
string ContentHtml	Content (see page 1198) HTML to display
string SearchText	Search (see page 899) Text
string SummaryHtml	Summary HTML
string ContentLanguage	Content (see page 1198) Language
long FolderId	Folder (see page 1236) to place the content in
object GoLive	Go Live Date
object EndDate	Content (see page 1198) End Date
string MetaInfoXml	Meta Info XML

Returns

Returns the numeric ID of newly created content.

Content.AddContent Method (string, string, string, string, string, string, long, object, object, string, long)

Adds Content (see page 1198) to the CMS

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContent With ContentID", Description = "Adds a piece of
content.")]
public long AddContent(string ContentTitle, string ContentComment, string ContentHtml, string
SearchText, string SummaryHtml, string ContentLanguage, long FolderId, object GoLive, object
EndDate, string MetaInfoXml, long ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContent With ContentID", Description = "Adds a piece of
content.")>
Public Function AddContent(ContentTitle As string, ContentComment As string, ContentHtml As
string, SearchText As string, SummaryHtml As string, ContentLanguage As string, FolderId As
long, GoLive As object, EndDate As object, MetaInfoXml As string, ContentID As long) As long
```

Parameters

Parameters	Description
string ContentTitle	Title of content to add
string ContentComment	Content (see page 1198) comments
string ContentHtml	Content (see page 1198) HTML to display
string SearchText	Search (see page 899) Text
string SummaryHtml	Summary HTML
string ContentLanguage	Content (see page 1198) Language
long FolderId	Folder (see page 1236) to place the content in
object GoLive	Go Live Date
object EndDate	Content (see page 1198) End Date
string MetaInfoXml	Meta Info XML
long ContentID	Content (see page 1198) ID

Returns

The numeric ID of newly created content.

Content.AddContent Method (string, string, string, string, string, string, long, object, object, string, long, long)

Adds Content (see page 1198) to the CMS

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContent with XmlID and TemplateID", Description = "Adds a piece
of content.")]
public long AddContent(string ContentTitle, string ContentComment, string ContentHtml, string
SearchText, string SummaryHtml, string ContentLanguage, long FolderId, object GoLive, object
EndDate, string MetaInfoXml, long XmlId, long TemplateID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContent with XmlID and TemplateID", Description = "Adds a piece
of content.")>
Public Function AddContent(ContentTitle As string, ContentComment As string, ContentHtml As
string, SearchText As string, SummaryHtml As string, ContentLanguage As string, FolderId As
long, GoLive As object, EndDate As object, MetaInfoXml As string, XmlId As long, TemplateID As
long) As long
```

Parameters

Parameters	Description
string ContentTitle	Title of content to add
string ContentComment	Content (see page 1198) comments
string ContentHtml	Content (see page 1198) HTML to display
string SearchText	Search (see page 899) Text
string SummaryHtml	Summary HTML
string ContentLanguage	Content (see page 1198) Language
long FolderId	Folder (see page 1236) to place the content in
object GoLive	Go Live Date
object EndDate	Content (see page 1198) End Date
long XmlId	XmlId
long TemplateID	TemplateID

Returns

The numeric ID of newly created content.

Content.AddContent Method (string, string, string, string, string, string, string, string, string, long, object, object, string)

Adds Content (see page 1198) to the CMS

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContentWithUser", Description = "Adds a piece of content.")]
public long AddContent(string Username, string Password, string Domain, string ContentTitle,
string ContentComment, string ContentHtml, string SearchText, string SummaryHtml, string
ContentLanguage, long FolderId, object GoLive, object EndDate, string MetaInfoXml);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContentWithUser", Description = "Adds a piece of content.")>
Public Function AddContent(Username As string, Password As string, Domain As string,
ContentTitle As string, ContentComment As string, ContentHtml As string, SearchText As string,
SummaryHtml As string, ContentLanguage As string, FolderId As long, GoLive As object, EndDate
As object, MetaInfoXml As string) As long
```

Parameters

Parameters	Description
string ContentTitle	Users (see page 1076) login name Domain name Title of content to add
string ContentComment	User (see page 1369) login password Content (see page 1198) comments
string ContentHtml	Content (see page 1198) HTML to display
string SearchText	Search (see page 899) Text
string SummaryHtml	Summary HTML
string ContentLanguage	Content (see page 1198) Language
long FolderId	Folder (see page 1236) to place the content in
object GoLive	Go Live Date
object EndDate	Content (see page 1198) End Date
string MetaInfoXml	Meta Info XML

Returns

Returns the numeric ID of newly created content.

Content.CheckOutContent Method

This will check the content out to the current user if the content is currently checked in and the user has privileges to check it out.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CheckOutContent", Description = "Checks the content out to the
current user")]
public bool CheckOutContent(long ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CheckOutContent", Description = "Checks the content out to the
current user")>
Public Function CheckOutContent(ContentID As long) As bool
```

Parameters

Parameters	Description
ContentId	ID of the content to check out

Returns

A Boolean representing the success of the check out

Content.CopyContentToFolder Method

Copy the content into required folder.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CopyContentToFolder", Description = "Copy the content into required
folder.")]
public ContentData CopyContentToFolder(string ContentID, long FolderID, string Language, bool
Publish);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CopyContentToFolder", Description = "Copy the content into required
folder.")>
Public Function CopyContentToFolder(ContentID As string, FolderID As long, Language As string,
Publish As bool) As ContentData
```

Parameters

Parameters	Description
Id	Content (see page 1198) ID
FID	Folder (see page 1236) ID
Lang	Language ID

Returns

Ektron.Cms.ContentData

Content.DeleteContentItem Method

Delete the content by the content's ID.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteContentItem", Description = "Delete the content by the
content's ID")]
public bool DeleteContentItem(long ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteContentItem", Description = "Delete the content by the
content's ID")>
Public Function DeleteContentItem(ContentID As long) As bool
```

Parameters

Parameters	Description
long ContentID	The content's numeric ID.

Content.GetAddViewLanguage Method

Loads all the languages from the system for a given content ID and populates the Add/View drop down box with flag set on each item.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAddViewLanguage", Description = "Gets all the languages from the
system for a given content ID ")]
public LanguageData[] GetAddViewLanguage(long ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAddViewLanguage", Description = "Gets all the languages from the
system for a given content ID ")>
Public Function GetAddViewLanguage(ContentID As long) As LanguageData()
```

Parameters

Parameters	Description
long ContentID	The Content (see page 1198)'s numeric ID.

Content.GetAllComments Method

Loads all of the comments.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllComments", Description = "Gets all comments")]
public CommentData[] GetAllComments(long KeyID, long CommentID, long RefID, string RefType,
long UserID, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllComments", Description = "Gets all comments")>
Public Function GetAllComments(KeyID As long, CommentID As long, RefID As long, RefType As
string, UserID As long, OrderBy As string) As CommentData()
```

Parameters

Parameters	Description
long KeyID	The numeric key ID.
long CommentID	The comment's numeric ID.
long RefID	Reference ID, (content or forms).
string RefType	Reference type, C = content or T = Task (see page 1085)
long UserID	The user's numeric ID.
string OrderBy	The order of the comments.

Content.GetAllTemplates Method

Loads all of the templates.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllTemplates", Description = "Gets all templates")]
```

```
public TemplateData[] GetAllTemplates(string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllTemplates", Description = "Gets all templates")>
Public Function GetAllTemplates(OrderBy As string) As TemplateData()
```

Parameters

Parameters	Description
string OrderBy	Order by clause

Content.GetAllUnassignedItemApprovals Method

Returns all of the unassigned user/groups for a given content ID.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUnassignedItemApprovals", Description = "Returns all of the
unassigned user/groups for a given content ID")]
public ApprovalData[] GetAllUnassignedItemApprovals(long ContentID, string ItemType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUnassignedItemApprovals", Description = "Returns all of the
unassigned user/groups for a given content ID")>
Public Function GetAllUnassignedItemApprovals(ContentID As long, ItemType As string) As
ApprovalData()
```

Parameters

Parameters	Description
long ContentID	The content's numeric ID.
string ItemType	The type of user.

GetChildContent Method

This is the overview for the GetChildContent method overload.

Overload List

	Name	Description
	Content.GetChildContent (long, bool, string) (see page 1206)	Loads all of the content for the given folder. This method can be accessed through the Content.asmx file located in:
	Content.GetChildContent (long, bool, string, PagingInfo) (see page 1207)	Loads all of the contents for the given folder. This method can be accessed through the Content.asmx file located in:

Content.GetChildContent Method (long, bool, string)

Loads all of the content for the given folder.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetChildContent", Description = "Loads all of the contents for the
given folder")]
public ContentData[] GetChildContent(long FolderID, bool Recursive, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetChildContent", Description = "Loads all of the contents for the
given folder")>
Public Function GetChildContent(FolderID As long, Recursive As bool, OrderBy As string) As
ContentData()
```

Parameters

Parameters	Description
long FolderID	The folder's numeric ID.
bool Recursive	True = recursive look up. False = none.
string OrderBy	An order by clause. Choices are: • **id** - the content ID. • **title** - the title of the content item. • **status** - whether the content is checked-in, published, checked-out, etc. • **language** - the language ID of the content. • **datecreated** - the date the content was created. • **datemodified** - the date the content was last modified. • **userid** - the ID of the user who last worked with the content.

Description

Content.GetChildContent Method (long, bool, string, PagingInfo)

Loads all of the contents for the given folder.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetChildContentWithPaging", Description = "Loads all of the contents
for the given folder")]
public ContentData[] GetChildContent(long FolderID, bool Recursive, string OrderBy, PagingInfo
pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetChildContentWithPaging", Description = "Loads all of the contents
for the given folder")>
Public Function GetChildContent(FolderID As long, Recursive As bool, OrderBy As string,
pagingInfo As PagingInfo) As ContentData()
```

Parameters

Parameters	Description
bool Recursive	True = recursive look up. False = none
string OrderBy	An order by clause. Choices are: • **id** - the content ID. • **title** - the title of the content item. • **status** - whether the content is checked-in, published, checked-out, etc. • **language** - the language ID of the content. • **datecreated** - the date the content was created. • **datemodified** - the date the content was last modified. • **userid** - the ID of the user who last worked with the content.
Id	the folders ID
pagignInfo	Paging Information

Returns

Ektron.Cms.ContentData()

Description

Content.GetContent Method

Get Content (see page 1198) By ID.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContent", Description = "Gets the content details by content id")]
public ContentData GetContent(long ContentID, Ektron.Cms.Content.EkContent.ContentResultType
ResultType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContent", Description = "Gets the content details by content id")>
Public Function GetContent(ContentID As long, ResultType As
Ektron.Cms.Content.EkContent.ContentResultType) As ContentData
```

Parameters

Parameters	Description
long ContentID	The content's numeric ID.
Preview	Preview
Published	Is Published

Content.GetContentByHistory Method

Retrieves the corresponding content by the history ID.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentByHistory", Description = "Retrieves the corresponding
content by the history ID")]
public ContentData GetContentByHistory(long HistoryID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentByHistory", Description = "Retrieves the corresponding
content by the history ID")>
Public Function GetContentByHistory(HistoryID As long) As ContentData
```

Parameters

Parameters	Description
long HistoryID	The history's numeric ID.

Content.GetContentForEditing Method

Loads the content details with checkout mode.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentForEditing", Description = "Loads the content details with
checkout mode")]
public ContentEditData GetContentForEditing(long ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentForEditing", Description = "Loads the content details with
checkout mode")>
Public Function GetContentForEditing(ContentID As long) As ContentEditData
```

Parameters

Parameters	Description
long ContentID	The content's numeric ID.

Content.GetContentState Method

Loads the content's details.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentState", Description = "Gets the content's details")]
public ContentStateData GetContentState(long ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentState", Description = "Gets the content's details")>
```

```
Public Function GetContentState(ContentID As long) As ContentStateData
```

Parameters

Parameters	Description
long ContentID	The content's numeric ID.

Content.GetContentStatus Method

This will return a single letter string which is the status of the piece of content. The letters returned are the same letters used in the Workarea view.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentStatus", Description = "Gets single letter string which is
the status of the piece of content")]
public string GetContentStatus(long ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentStatus", Description = "Gets single letter string which is
the status of the piece of content")>
Public Function GetContentStatus(ContentID As long) As string
```

Parameters

Parameters	Description
long ContentID	The numeric ID of the content for which to return a status.

Content.GetDomain Method

Gets web domain for a piece of content.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDomain", Description = "Gets web domain for a piece of content")]
public string GetDomain(long ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDomain", Description = "Gets web domain for a piece of content")>
Public Function GetDomain(ContentID As long) As string
```

Parameters

Parameters	Description
long ContentID	The content's ID

Content.MoveContentToFolder Method

Moves content to a folder.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MoveContentToFolder", Description = "Moves content to a folder")]
public ContentData MoveContentToFolder(string ContentID, string ContentLanguage, long
FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MoveContentToFolder", Description = "Moves content to a folder")>
Public Function MoveContentToFolder(ContentID As string, ContentLanguage As string, FolderID
As long) As ContentData
```

Parameters

Parameters	Description
string ContentID	The content's ID.
string ContentLanguage	The content's language.
long FolderID	The folder's numeric ID.

Content.PublishContent Method

Function takes in information about the content to be checked in and published. This goes through the normal approval chain. The content must be in a checked out state to the current logged in user for this to succeed.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "PublishContent", Description = "Function takes in information about
the content to be checked in and published. This goes through the normal approval chain. The
content must be in a checked out state to the current logged in user for this to succeed")]
public bool PublishContent(long ContentID, long FolderID, int ContentLanguage, string
DontCreateTask, long UserID, string TaskTitle);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "PublishContent", Description = "Function takes in information about
the content to be checked in and published. This goes through the normal approval chain. The
content must be in a checked out state to the current logged in user for this to succeed")>
Public Function PublishContent(ContentID As long, FolderID As long, ContentLanguage As
Integer, DontCreateTask As string, UserID As long, TaskTitle As string) As bool
```

Parameters

Parameters	Description
long ContentID	The numeric value of the content's ID.
long FolderID	The numeric value of the content folder's ID.
int ContentLanguage	The numeric value of the content's language ID.
string DontCreateTask	If this parameter is "", a task will be created for this submit. Otherwise, no task is created.
long UserID	The user ID of task assigned by current user. This is needed only if "dontCreateTask" is "", otherwise it is ignored.
string TaskTitle	Title for the task. This is needed only if "dontCreateTask" is "", otherwise it is ignored.

Content.RenameContent Method

Renames the content.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RenameContent", Description = "Renames the content by ID")]
public ContentData RenameContent(long ContentID, string Title);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RenameContent", Description = "Renames the content by ID")>
Public Function RenameContent(ContentID As long, Title As string) As ContentData
```

Parameters

Parameters	Description
long ContentID	The content's numeric ID.
string Title	The content's new title.

Content.SaveContent Method

Updates the checked out content block. Only following properties can be updated:

- contentData.Title
- contentData.Teaser
- contentData.Comment
- contentData.GoLive
- contentData.Html
- contentData.EndDate
- contentData.EndDateAction
- contentData.ManualAlias

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SaveContent", Description = "Updates the checked out content
block.")]
public bool SaveContent(ContentEditData contentEditData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SaveContent", Description = "Updates the checked out content
block.")>
Public Function SaveContent(contentEditData As ContentEditData) As bool
```

Parameters

Parameters	Description
contentData	

Returns

False if successful

Remarks

Content (see page 1198) must be in the checkedout state.

Content.SubmitForDelete Method

Delete the content by the content's ID and folder ID.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SubmitForDelete", Description = "Deletes the content by the
content's Id and folder Id")]
public bool SubmitForDelete(long ContentID, long FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SubmitForDelete", Description = "Deletes the content by the
content's Id and folder Id")>
Public Function SubmitForDelete(ContentID As long, FolderID As long) As bool
```

Parameters

Parameters	Description
long ContentID	The content's numeric ID.
long FolderID	The content's numeric folder ID.

Content.UndoCheckout Method

This undoes the checkout for a piece of content which is checked out to the current user.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UndoCheckout", Description = "Reverts the checkout for a piece of
content which is checked out to the current user")]
public bool UndoCheckout(long ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UndoCheckout", Description = "Reverts the checkout for a piece of
content which is checked out to the current user")>
Public Function UndoCheckout(ContentID As long) As bool
```

Parameters

Parameters	Description
long ContentID	The numeric ID of the content for which to undo check out.

Content.UpdateContentMetaData Method

Updates a specific metadata item attached to a specified piece of content.

This method can be accessed through the Content.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Content.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateContentMetaData", Description = "Updates Content Metadata")]
public bool UpdateContentMetaData(long ContentID, long MetaDataTypeID, string MetaDataText);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateContentMetaData", Description = "Updates Content Metadata")>
Public Function UpdateContentMetaData(ContentID As long, MetaDataTypeID As long, MetaDataText
As string) As bool
```

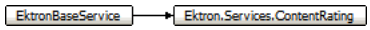
ContentRating Class

This section describes the Web Service methods used with Ektron's Content (see page 1198) Ratings. The Content (see page 1198) Rating Web service is a wrapper for the Content (see page 1198) Rating API (see page 669): `Ektron.Cms.API.Content.ContentRating` (see page 866). These methods can be accessed through the ContentRating.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ContentRating.asmx
```

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class ContentRating : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class ContentRating
Inherits EktronBaseService
```

File

ContentRating.cs

ContentRating Methods

The methods of the ContentRating class are listed here.

Public Methods

	Name	Description
	AddContentRating (see page 1215)	Adds a content rating to the database. Internally a check is done to ensure the user has not already rated the content, and passes down the rating and any comments the user supplied. This method can be accessed through the ContentRating.asmx file located in:
	GetContentRatingResults (see page 1215)	Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content from the range of StartDate to EndDate. In order to get the full range of ratings, pass DateTime.Min and DateTime.Max as arguments. This method can be accessed through the ContentRating.asmx file located in:
	PurgeContentRatings (see page 1216)	Purges all ratings of a particular content block over a defined date/time range. This method can be accessed through the ContentRating.asmx file located in:

ContentRating.AddContentRating Method

Adds a content rating to the database. Internally a check is done to ensure the user has not already rated the content, and passes down the rating and any comments the user supplied.

This method can be accessed through the ContentRating.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ContentRating.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContentRating", Description = "")]
public void AddContentRating(long contentid, long userid, int userRating, string userComments);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContentRating", Description = "")>
Public Function AddContentRating(contentid As long, userid As long, userRating As Integer,
userComments As string) As void
```

Parameters

Parameters	Description
long contentid	The numeric ID associated with the content block being rated.
long userid	The unique global identifier stored as a cookie by the user.
int userRating	The rating the user supplies (1-9)
string userComments	Optional comments that the user provides.

ContentRating.GetContentRatingResults Method

Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content from the range of StartDate to EndDate. In order to get the full range of ratings, pass DateTime.Min and DateTime.Max as arguments.

This method can be accessed through the ContentRating.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ContentRating.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentRatingResults", Description = "")]
public System.Data.DataSet GetContentRatingResults(long contentID, DateTime startDate,
DateTime endDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentRatingResults", Description = "")>
Public Function GetContentRatingResults(contentID As long, startDate As DateTime, endDate As
DateTime) As System.Data.DataSet
```

Parameters

Parameters	Description
long contentID	The numeric ID associated with the content block whose ratings are being queried.
DateTime startDate	The start date of a range of dates to query over.
DateTime endDate	The end date of a range of dates to query over.

Returns

(Username - String, RatingDate - DateTime, Rating - int, Comments - String)

ContentRating.PurgeContentRatings Method

Purges all ratings of a particular content block over a defined date/time range.

This method can be accessed through the ContentRating.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/ContentRating.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "PurgeContentRatings", Description = "")]
public void PurgeContentRatings(long contentid, DateTime startDate, DateTime endDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "PurgeContentRatings", Description = "")>
Public Function PurgeContentRatings(contentid As long, startDate As DateTime, endDate As
DateTime) As void
```

Parameters

Parameters	Description
long contentid	The ID of the content that is having its ratings purged.
DateTime startDate	The beginning of the date range of content to be purged.
DateTime endDate	The end of the date range of content to be purged.

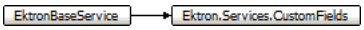
CustomFields Class

This section describes the Web Service methods used with Ektron's Custom Fields. The Custom Fields Web service is a wrapper for the Custom Fields API (see page 669): `Ektron.Cms.API.CustomFields` (see page 673). These methods can be accessed through the CustomFields.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/CustomFields.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class CustomFields : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class CustomFields
Inherits EktronBaseService
```

File

CustomFields.cs

CustomFields Methods

The methods of the CustomFields class are listed here.

Public Methods

	Name	Description
	AddNewEditableCustomFieldAssignments (see page 1218)	Returns the HTML string for the add/edit custom fields. This method can be accessed through the CustomFields.asmx file located in:
	BoolToYesNo (see page 1218)	Returns labels for yes or no based on locale. This method can be accessed through the CustomFields.asmx file located in:
	GetEditableCustomFieldAssignments (see page 1219)	Returns the editable custom fields. Break Inheritance check box: • Visible if not at root. • Checked if any fields assigned without inheritance. • If unchecked & form saved, removes all assignments. • If checked & form saved and no assignments chosen, creates NULL assignment (CustomFieldID -1). • When checked, enables all assignment check boxes (iterate through all of this tables elements). • When unchecked, disables all assignment check boxes (iterate through all of this tables elements). • Assignment Check boxes: • When clicked: if 'Assignment' button is unchecked, then ensure required us unchecked. • When clicked: if 'Required' button, don't accept checks if assigned is unchecked. • When form... more (see page 1219)
	GetEnhancedMetadataArea (see page 1220)	Returns the HTML needed to render the EnhancedMetadata selection area. This method can be accessed through the CustomFields.asmx file located in:
	GetEnhancedMetadataScript (see page 1220)	Returns support Javascript code for enhanced metadata. This method can be accessed through the CustomFields.asmx file located in:
	GetFieldsByFolder (see page 1220)	Gets the custom fields associated with a folder by supplying the Folder (see page 1236) ID and Language ID. This method can be accessed through the CustomFields.asmx file located in:
	GetSearchProperties (see page 1221)	Returns an HTML string for the search properties display. This method can be accessed through the CustomFields.asmx file located in:
	ProcessCustomFields (see page 1221)	Process Folders' Custom-Field (and metadata) assignments: When form submitted, copy checked items to hidden field. Form (see page 1247) processor: • If break inheritance checked, and any items in hidden field, send contents to db. • If break inheritance checked, and hidden field empty, create dummy assignment fieldID = -1. • If break inheritance NOT checked, remove all assignments for this folder. This method can be accessed through the CustomFields.asmx file located in:
	SearchAllAssets (see page 1222)	Returns the search result based on assets. This method can be accessed through the CustomFields.asmx file located in:
	WriteFilteredMetadataForEdit (see page 1222)	Returns an HTML string for filtered metadata. This method can be accessed through the CustomFields.asmx file located in:
	WriteFilteredMetadataForView (see page 1223)	Returns an HTML string for metadata view. This method can be accessed through the CustomFields.asmx file located in:

	WriteLibrarySearchExtended (see page 1223)	Returns an HTML string for the library search custom metadata properties. This method can be accessed through the CustomFields.asmx file located in:
	WriteMetadataDefaultForEdit (see page 1224)	Returns an HTML string for the default metadata. This method can be accessed through the CustomFields.asmx file located in:
	WriteMetadataTypeForView (see page 1224)	Returns a metadata type for view. This method can be accessed through the CustomFields.asmx file located in:
	WritePublicSearch (see page 1225)	Returns WritepublicsearchEx. This method can be accessed through the CustomFields.asmx file located in:

CustomFields.AddNewEditableCustomFieldAssignments Method

Returns the HTML string for the add/edit custom fields.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddNewEditableCustomFieldAssignments", Description = "")]
public string AddNewEditableCustomFieldAssignments(long parentfolderid);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddNewEditableCustomFieldAssignments", Description = "")>
Public Function AddNewEditableCustomFieldAssignments(parentfolderid As long) As string
```

Parameters

Parameters	Description
long parentfolderid	The parent folder's ID.

Returns

String

CustomFields.BoolToYesNo Method

Returns labels for yes or no based on locale.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "BoolToYesNo", Description = "")]
public string BoolToYesNo(bool Value);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "BoolToYesNo", Description = "")>
Public Function BoolToYesNo(Value As bool) As string
```

Parameters

Parameters	Description
bool Value	True or False

Returns

String

CustomFields.GetEditableCustomFieldAssignments Method

Returns the editable custom fields.

Break Inheritance check box:

- Visible if not at root.
- Checked if any fields assigned without inheritance.
- If unchecked & form saved, removes all assignments.
- If checked & form saved and no assignments chosen, creates NULL assignment (CustomFieldID -1).
- When checked, enables all assignment check boxes (iterate through all of this tables elements).
- When unchecked, disables all assignment check boxes (iterate through all of this tables elements).
- Assignment Check boxes:
- When clicked: if 'Assignment' button is unchecked, then ensure required us unchecked.
- When clicked: if 'Required' button, don't accept checks if assigned is unchecked.
- When form submitted, copy checked items to hidden field.
- Form (see page 1247) processor;
- If break inheritance checked, and any items in hidden field, send contents to db.
- If break inheritance checked, and hidden field empty, create dummy assignment fieldID = -1.
- If break inheritance NOT checked, remove all assignments for this folder.

Note: If editMode is false, then items are visible but not changeable by user.

This method can be accessed through the CustomFields.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/CustomFields.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetEditableCustomFieldAssignments", Description = "")]
public string GetEditableCustomFieldAssignments(object folderID, object editMode);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetEditableCustomFieldAssignments", Description = "")>
Public Function GetEditableCustomFieldAssignments(folderID As object, editMode As object) As
string
```

Parameters

Parameters	Description
object folderID	The folder's ID.
object editMode	Edit mode.

Returns

String

Remarks

Function GetEditableCustomFieldAssignments: "

CustomFields.GetEnhancedMetadataArea Method

Returns the HTML needed to render the EnhancedMetadata selection area.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetEnhancedMetadataArea", Description = "")]
public string GetEnhancedMetadataArea();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetEnhancedMetadataArea", Description = "")>
Public Function GetEnhancedMetadataArea() As string
```

CustomFields.GetEnhancedMetadataScript Method

Returns support Javascript code for enhanced metadata.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetEnhancedMetadataScript", Description = "")]
public string GetEnhancedMetadataScript();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetEnhancedMetadataScript", Description = "")>
Public Function GetEnhancedMetadataScript() As string
```

Returns

String

CustomFields.GetFieldsByFolder Method

Gets the custom fields associated with a folder by supplying the Folder (see page 1236) ID and Language ID.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
public Collection GetFieldsByFolder(long FolderID, int LanguageID);
```

Visual Basic

```
Public Function GetFieldsByFolder(FolderID As long, LanguageID As Integer) As Collection
```

Parameters

Parameters	Description
long FolderID	The folder's ID.
int LanguageID	The language's ID.

Returns

Microsoft.VisualBasic.Collection (see page 95)

CustomFields.GetSearchProperties Method

Returns an HTML string for the search properties display.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetSearchProperties", Description = "")]
public StringBuilder GetSearchProperties(SearchAssetDispayRequest sadReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetSearchProperties", Description = "")>
Public Function GetSearchProperties(sadReq As SearchAssetDispayRequest) As StringBuilder
```

Parameters

Parameters	Description
SearchAssetDispayRequest sadReq	The search asset objects. (Required)

Returns

System.Text.StringBuilder

CustomFields.ProcessCustomFields Method

Process Folders' Custom-Field (and metadata) assignments:

When form submitted, copy checked items to hidden field.

Form (see page 1247) processor:

- If break inheritance checked, and any items in hidden field, send contents to db.
- If break inheritance checked, and hidden field empty, create dummy assignment fieldID = -1.
- If break inheritance NOT checked, remove all assignments for this folder.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ProcessCustomFields", Description = "")]
public object ProcessCustomFields(long folderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ProcessCustomFields", Description = "")>
Public Function ProcessCustomFields(folderID As long) As object
```

Parameters

Parameters	Description
long folderID	The folder's ID.

Returns

Object

CustomFields.SearchAllAssets Method

Returns the search result based on assets.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SearchAllAssets", Description = "")]
public StringBuilder SearchAllAssets(SearchAssetDispayRequest sadReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SearchAllAssets", Description = "")>
Public Function SearchAllAssets(sadReq As SearchAssetDispayRequest) As StringBuilder
```

Parameters

Parameters	Description
SearchAssetDispayRequest sadReq	Search (see page 899) parameters. (Required)

Returns

System.Text.StringBuilder

CustomFields.WriteFilteredMetadataForEdit Method

Returns an HTML string for filtered metadata.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
public StringBuilder WriteFilteredMetadataForEdit(ContentMetaData[] MetadataTypes, bool
SearchableOnly, string ty, long FolderId, int Count, Hashtable metaPermsObj);
```

Visual Basic

```
Public Function WriteFilteredMetadataForEdit(MetadataTypes As ContentMetaData[],
SearchableOnly As bool, ty As string, FolderId As long, Count As Integer, metaPermsObj As
Hashtable) As StringBuilder
```

Parameters

Parameters	Description
ContentMetaData[] MetadataTypes	Metadata (see page 1300) types.
bool SearchableOnly	bSearchableOnly
string ty	type
long FolderId	The folder's ID.
int Count	Number of items. (Required)

Returns

System.Text.StringBuilder

Description

[WebMethod(MessageName = "WriteFilteredMetadataForEdit", Description = "")]

CustomFields.WriteFilteredMetadataForView Method

Returns an HTML string for metadata view.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "WriteFilteredMetadataForView", Description = "")]
public string WriteFilteredMetadataForView(ContentMetaData[] MetadataTypes, long FolderId,
bool SearchableOnly);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "WriteFilteredMetadataForView", Description = "")>
Public Function WriteFilteredMetadataForView(MetadataTypes As ContentMetaData[], FolderId As
long, SearchableOnly As bool) As string
```

Parameters

Parameters	Description
ContentMetaData[] MetadataTypes	Metadata (see page 1300) types.
long FolderId	The folder's ID.
bool SearchableOnly	True or False

Returns

String

CustomFields.WriteLibrarySearchExtended Method

Returns an HTML string for the library search custom metadata properties.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "WriteLibrarySearchExtended", Description = "")]
public StringBuilder WriteLibrarySearchExtended(long folderId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "WriteLibrarySearchExtended", Description = "")>
Public Function WriteLibrarySearchExtended(folderId As long) As StringBuilder
```

Parameters

Parameters	Description
long folderId	The folder's ID.

Returns

System.Text.StringBuilder

CustomFields.WriteMetadataDefaultForEdit Method

Returns an HTML string for the default metadata.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
public StringBuilder WriteMetadataDefaultForEdit(ContentMetaData MetadataType, int DataType,
Hashtable metaPermsObj);
```

Visual Basic

```
Public Function WriteMetadataDefaultForEdit(MetadataType As ContentMetaData, DataType As
Integer, metaPermsObj As Hashtable) As StringBuilder
```

Parameters

Parameters	Description
int DataType	Data types
cMetadataType	Metadata (see page 1300) values.

Returns

System.Text.StringBuilder

Description

[WebMethod(MessageName = "WriteMetadataDefaultForEdit", Description = "")]

CustomFields.WriteMetadataTypeForView Method

Returns a metadata type for view.

This method can be accessed through the CustomFields.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/CustomFields.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "WriteMetadataTypeForView", Description = "")]
public string WriteMetadataTypeForView(string DataType, string Caption, string Value);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "WriteMetadataTypeForView", Description = "")>
Public Function WriteMetadataTypeForView(DataType As string, Caption As string, Value As
string) As string
```

Parameters

Parameters	Description
string DataType	Data types.
string Caption	Caption value.

string Value	Value. (Required)

Returns

String

CustomFields.WritePublicSearch Method

Returns WritepublicsearchEx.

This method can be accessed through the CustomFields.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/CustomFields.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "WritePublicSearch", Description = "")]
public StringBuilder WritePublicSearch(SearchAssetDispayRequest sadReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "WritePublicSearch", Description = "")>
Public Function WritePublicSearch(sadReq As SearchAssetDispayRequest) As StringBuilder
```

Parameters

Parameters	Description
SearchAssetDispayRequest sadReq	Search (see page 899) asset data. (Required)

Returns

System.Text.StringBuilder

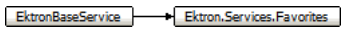
Favorites Class

This section describes the Web Service methods used with Ektron's Favorites feature. This feature is a part Ektron's Community (see page 750) Platform. The Favorites Web service is a wrapper for the Favorites API (see page 669): `Ektron.Cms.API.Community.Favorites` (see page 763). These methods can be accessed through the Favorites.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Community/Favorites.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
public class Favorites : EktronBaseService;
```

Visual Basic

```
Public Class Favorites
Inherits EktronBaseService
```

File

Favorites.cs

Favorites Methods

The methods of the Favorites class are listed here.

Public Methods

	Name	Description
	AddContentFavorite (see page 1226)	This is the overview for the AddContentFavorite method overload.
	DeleteContentFavorite (see page 1227)	Deletes a piece of content from a user's favorites. This method can be accessed through the Favorites.asmx file located in:
	GetContentFavorites (see page 1228)	This is the overview for the GetContentFavorites method overload.
	GetFavoritesDirectory (see page 1230)	Returns a user's Favorites (see page 1225) taxonomy directory structure. This method can be accessed through the Favorites.asmx file located in:
	IsContentFavorite (see page 1230)	Checks if a content item is in a user's favorites. This method can be accessed through the Favorites.asmx file located in:
	MoveContentFavorite (see page 1231)	Moves a content favorite to a new favorites directory. This method can be accessed through the Favorites.asmx file located in:

AddContentFavorite Method

This is the overview for the AddContentFavorite method overload.

Description

This method is overloaded.

- AddContentFavorite (Long, Long) - adds a piece of content to user's favorites.
- AddContentFavorite (Long, Long, Long) - adds a piece of content to user's favorites and places it in a subdirectory.

Overload List

	Name	Description
	Favorites.AddContentFavorite (long, long) (see page 1226)	Adds a piece of content to user's favorites. This method can be accessed through the Favorites.asmx file located in:
	Favorites.AddContentFavorite (long, long, long) (see page 1227)	Adds a piece of content to user's favorites and places it in a subdirectory. This method can be accessed through the Favorites.asmx file located in:

Favorites.AddContentFavorite Method (long, long)

Adds a piece of content to user's favorites.

This method can be accessed through the Favorites.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Favorites.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContentFavorite", Description = "Adds a piece of content to
user's favorites.")]
public void AddContentFavorite(long contentId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "AddContentFavorite", Description = "Adds a piece of content to
user's favorites.")>
Public Function AddContentFavorite(contentId As long, userId As long) As void
```

Parameters

Parameters	Description
long contentId	ID of the content item.
long userId	ID of the user.

Favorites.AddContentFavorite Method (long, long, long)

Adds a piece of content to user's favorites and places it in a subdirectory.

This method can be accessed through the Favorites.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Favorites.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContentFavorite to directory", Description = "Adds a content item
to a user's favorites and places it in a subdirectory.")]
public void AddContentFavorite(long contentId, long userId, long directoryId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContentFavorite to directory", Description = "Adds a content item
to a user's favorites and places it in a subdirectory.")>
Public Function AddContentFavorite(contentId As long, userId As long, directoryId As long) As
void
```

Parameters

Parameters	Description
long contentId	ID of the content item.
long userId	ID of the user.
long directoryId	ID of the directory to add the content in the user's favorites.

Favorites.DeleteContentFavorite Method

Deletes a piece of content from a user's favorites.

This method can be accessed through the Favorites.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Favorites.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteContentFavorite", Description = "Deletes a content item from a
user's favorites.")]
public void DeleteContentFavorite(long contentId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteContentFavorite", Description = "Deletes a content item from a
user's favorites.")>
Public Function DeleteContentFavorite(contentId As long, userId As long) As void
```

Parameters

Parameters	Description
long contentId	ID of the content item.
long userId	ID of the user.

GetContentFavorites Method

This is the overview for the GetContentFavorites method overload.

Description

This method is overloaded.

- GetContentFavorites (Long, PagingInfo) - returns a list of the user's favorites by providing the user's ID. This method allows you to page the favorites.
- GetContentFavorites (Long, Long) - returns a list of the user's favorites in a directory by providing the user's ID and the ID of the directory.
- GetContentFavorites (Long, Long, PagingInfo) - returns a list of the user's favorites in a directory by providing the user's ID and the ID of the directory. This method allows you to page the favorites.

Overload List

	Name	Description
	Favorites.GetContentFavorites (long, long) (see page 1229)	Returns a list of the user's favorites in a directory. This method can be accessed through the Favorites.asmx file located in:
	Favorites.GetContentFavorites (long, long, PagingInfo) (see page 1229)	Returns a list of the user's favorites in a directory. This method can be accessed through the Favorites.asmx file located in:
	Favorites.GetContentFavorites (long, PagingInfo) (see page 1228)	Returns a list of the user's favorites. This method can be accessed through the Favorites.asmx file located in:

Favorites.GetContentFavorites Method (long, PagingInfo)

Returns a list of the user's favorites.

This method can be accessed through the Favorites.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Favorites.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentFavorites with paging info", Description = "Returns a list
of the user's favorites.")]
public DirectoryContentData[] GetContentFavorites(long userId, PagingInfo pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentFavorites with paging info", Description = "Returns a list
of the user's favorites.")>
Public Function GetContentFavorites(userId As long, pagingInfo As PagingInfo) As
DirectoryContentData()
```

Parameters

Parameters	Description
PagingInfo pagingInfo	Paging information for limiting the resultset.
userID	ID of user to retrieve favorites for.

Returns

An array of DirectoryContentData objects.

Favorites.GetContentFavorites Method (long, long)

Returns a list of the user's favorites in a directory.

This method can be accessed through the Favorites.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Favorites.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentFavorites", Description = "Returns a list of the user's
favorites from a specified directory.")]
public DirectoryContentData[] GetContentFavorites(long userId, long directoryId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentFavorites", Description = "Returns a list of the user's
favorites from a specified directory.")>
Public Function GetContentFavorites(userId As long, directoryId As long) As
DirectoryContentData()
```

Parameters

Parameters	Description
long directoryId	ID of the directory to retrieve favorites from.
userID	ID of the user to retrieve favorites for.

Returns

An array of DirectoryContentData objects.

Favorites.GetContentFavorites Method (long, long, PagingInfo)

Returns a list of the user's favorites in a directory.

This method can be accessed through the Favorites.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Favorites.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentFavorites for directory with paging info", Description =
"Returns a list of the user's favorites in a directory.")]
public DirectoryContentData[] GetContentFavorites(long userId, long directoryId, PagingInfo
pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentFavorites for directory with paging info", Description =
"Returns a list of the user's favorites in a directory.")>
```

```
Public Function GetContentFavorites(userId As long, directoryId As long, pagingInfo As
PagingInfo) As DirectoryContentData()
```

Parameters

Parameters	Description
long directoryId	ID of directory to retrieve favorites from.
PagingInfo pagingInfo	Paging information for limiting the resultset.
userID	ID of user to retrieve favorites for.

Returns

An array of DirectoryContentData objects.

Favorites.GetFavoritesDirectory Method

Returns a user's Favorites (see page 1225) taxonomy directory structure.

This method can be accessed through the Favorites.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Favorites.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFavoritesDirectory", Description = "Returns user's Favorites
taxonomy directory structure.")]
public TaxonomyData GetFavoritesDirectory(long userId, long directoryId, PagingInfo
pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFavoritesDirectory", Description = "Returns user's Favorites
taxonomy directory structure.")>
Public Function GetFavoritesDirectory(userId As long, directoryId As long, pagingInfo As
PagingInfo) As TaxonomyData
```

Parameters

Parameters	Description
long userId	ID of the user whose favorites should be retrieved.
long directoryId	ID of the favorites directory to retrieve.
PagingInfo pagingInfo	Paging information for limiting the resultset.

Favorites.IsContentFavorite Method

Checks if a content item is in a user's favorites.

This method can be accessed through the Favorites.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Favorites.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsContentFavorite", Description = "Checks if a content item is in a
user's favorites.")]
public bool IsContentFavorite(long contentId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsContentFavorite", Description = "Checks if a content item is in a
user's favorites.")>
```

```
Public Function IsContentFavorite(contentId As long, userId As long) As bool
```

Parameters

Parameters	Description
long contentId	ID of the content item.
long userId	ID of the user.

Returns

True if the content item is in user's favorites, else false.

Favorites.MoveContentFavorite Method

Moves a content favorite to a new favorites directory.

This method can be accessed through the Favorites.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Favorites.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MoveContentFavorite", Description = "Moves a content favorite to a
new favorites directory.")]
public void MoveContentFavorite(long contentId, long userId, long directoryId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MoveContentFavorite", Description = "Moves a content favorite to a
new favorites directory.")>
Public Function MoveContentFavorite(contentId As long, userId As long, directoryId As long) As
void
```

Parameters

Parameters	Description
long contentId	ID of the content item to move.
long userId	ID of the user.
long directoryId	ID of the favorite directory where the content is being moved.

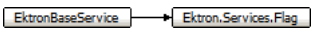
Flag Class

This section describes the Web Service methods used with Ektron's Flagging (see page 771) feature. This feature is a part Ektron's Community (see page 750) Platform. The Flagging (see page 771) Web service is a wrapper for the Flagging (see page 771) API (see page 669): `Ektron.Cms.API.Community.Flagging (see page 771)`. These methods can be accessed through the Flag.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Flag.asmx
```

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
public class Flag : EktronBaseService;
```

Visual Basic

```
Public Class Flag
```

```
Inherits EktronBaseService
```

File

Flag.cs

Description

[WebService(Namespace = "http://tempuri.org/")] [WebServiceBinding(ConformsTo = WsiProfiles.None)]

Flag Methods

The methods of the Flag class are listed here.

Public Methods

	Name	Description
	AddFlagObject (see page 1232)	Saves a new flag on an object. This method can be accessed through the Flagging.asmx file located in:
	AssignFlagToContent (see page 1233)	Assigns a flag definition to a piece of content. This method can be accessed through the Flagging.asmx file located in:
	GetAllFlagEntries (see page 1233)	Gets a list of flag entries for a piece of content over a given time period. This method can be accessed through the Flagging.asmx file located in:
	GetContentFlagData (see page 1234)	Returns flagData for a piece of content for a given user. This method can be accessed through the Flagging.asmx file located in:
	GetFlagDataById (see page 1234)	Retrieves a flag entry on a piece of content. This method can be accessed through the Flagging.asmx file located in:
	GetFlaggedRank (see page 1235)	Retrieves a flag ranking for an object. This method can be accessed through the Flagging.asmx file located in:
	PurgeFlagEntries (see page 1235)	Purges all flags of a particular content block over a defined date/time range. This method can be accessed through the Flagging.asmx file located in:
	RemoveObjectObject (see page 1235)	Removes a flag entry from an object. This method can be accessed through the Flagging.asmx file located in:
	UpdateFlagObject (see page 1236)	Updates an existing Flag (see page 1231) on an Object. This method can be accessed through the Flagging.asmx file located in:

Flag.AddFlagObject Method

Saves a new flag on an object.

This method can be accessed through the Flagging.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Flag.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddFlagObject", Description = "adds new flagged entry")]
public long AddFlagObject(ObjectFlagData objectFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddFlagObject", Description = "adds new flagged entry")>
Public Function AddFlagObject(objectFlag As ObjectFlagData) As long
```

Parameters

Parameters	Description
ObjectFlagData objectFlag	Object flag data to save.

Flag.AssignFlagToContent Method

Assigns a flag definition to a piece of content.

This method can be accessed through the Flagging.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Flag.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignFlagToContent", Description = "assign flag definition to
content")]
public void AssignFlagToContent(long contentId, long flagId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignFlagToContent", Description = "assign flag definition to
content")>
Public Function AssignFlagToContent(contentId As long, flagId As long) As void
```

Parameters

Parameters	Description
long contentId	ID of the content to be assigned the flag.
long flagId	ID of the flag definition.

Flag.GetAllFlagEntries Method

Gets a list of flag entries for a piece of content over a given time period.

This method can be accessed through the Flagging.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Flag.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllFlagEntries", Description = "Gets all flagged entries")]
public ContentFlagData[] GetAllFlagEntries(long contentId, int maxResults, DateTime startDate,
DateTime endDate, ref int totalFlags);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllFlagEntries", Description = "Gets all flagged entries")>
Public Function GetAllFlagEntries(contentId As long, maxResults As Integer, startDate As
DateTime, endDate As DateTime, ByRef totalFlags As Integer) As ContentFlagData()
```

Parameters

Parameters	Description
long contentId	ID of content item for which to retrieve flags.
int maxResults	Max number of results to retrieve.
DateTime startDate	Date to begin retrieving flags.
DateTime endDate	Date to end retrieving flags.
ref int totalFlags	Returns the total number of flags found.

Returns

Array of content flag data.

Flag.GetContentFlagData Method

Returns flagData for a piece of content for a given user.

This method can be accessed through the Flagging.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Flag.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentFlagData", Description = "gets flagged data using user or
visitor id")]
public ContentFlagData GetContentFlagData(long contentId, long userId, String visitorId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentFlagData", Description = "gets flagged data using user or
visitor id")>
Public Function GetContentFlagData(contentId As long, userId As long, visitorId As String) As
ContentFlagData
```

Parameters

Parameters	Description
contentid	Id of content to retrieve flag for.
userid	Id of cms user who flagged content.

Returns

ContentFlagData object.

Flag.GetFlagDataById Method

Retrieves a flag entry on a piece of content.

This method can be accessed through the Flagging.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Flag.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFlagDataById", Description = "gets content flag date using flag
id")]
public ContentFlagData GetFlagDataById(long flagId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFlagDataById", Description = "gets content flag date using flag
id")>
Public Function GetFlagDataById(flagId As long) As ContentFlagData
```

Parameters

Parameters	Description
long flagId	ID of the flag entry.

Returns

ContentFlagData

Flag.GetFlaggedRank Method

Retrieves a flag ranking for an object.

This method can be accessed through the Flagging.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Flag.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFlaggedRank", Description = "gets content by ranking")]
public FlaggedRankData[] GetFlaggedRank(long objectId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFlaggedRank", Description = "gets content by ranking")>
Public Function GetFlaggedRank(objectId As long) As FlaggedRankData()
```

Parameters

Parameters	Description
long objectId	ID of the object for which to retrieve a flag ranking.

Flag.PurgeFlagEntries Method

Purges all flags of a particular content block over a defined date/time range.

This method can be accessed through the Flagging.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Flag.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "PurgeFlagEntries", Description = "delets matched entries")]
public void PurgeFlagEntries(long objectId, DateTime startDate, DateTime endDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "PurgeFlagEntries", Description = "delets matched entries")>
Public Function PurgeFlagEntries(objectId As long, startDate As DateTime, endDate As DateTime)
As void
```

Parameters

Parameters	Description
long objectId	The ID of the content that is having its flags purged.
DateTime startDate	The beginning of the date range of content to be purged.
DateTime endDate	The end of the date range of content to be purged.

Flag.RemoveObjectObject Method

Removes a flag entry from an object.

This method can be accessed through the Flagging.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Flag.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "RemoveObjectObject", Description = "remove the flag definition")]
public void RemoveObjectObject(long flagId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RemoveObjectObject", Description = "remove the flag definition")>
Public Function RemoveObjectObject(flagId As long) As void
```

Parameters

Parameters	Description
long flagId	ID of flag entry to remove.

Flag.UpdateFlagObject Method

Updates an existing Flag (see page 1231) on an Object.

This method can be accessed through the Flagging.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Community/Flag.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateFlagObject", Description = "update flag definition")]
public void UpdateFlagObject(ObjectFlagData objectFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateFlagObject", Description = "update flag definition")>
Public Function UpdateFlagObject(objectFlag As ObjectFlagData) As void
```

Parameters

Parameters	Description
request	Request criteria defining the flag data for the object.

Folder Class

This section describes the Web Service methods used with folder. The Folder Web service is a wrapper for the Folder API (see page 669): `Ektron.Cms.API.Folder` (see page 683). These methods can be accessed through the Folder.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Folder.asmx`

Web Services (see page 1083) examples involving Folder are available in the Developer sample site. You can download this site from:

http://www.ektron.com/cms400-web-cms.aspx?id=5278

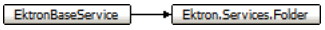
Once the developer site is installed, example code files are located in your site root. For example:

`C:\Inetpub\wwwroot\CMS400Developer\Developer\dotNetAssemblies\WebServicesAPI\FolderAPI.`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header. For example:

```
folderApi.AuthenticationHeaderValue = New localhost.AuthenticationHeader()
folderApi.AuthenticationHeaderValue.Username = "admin"
folderApi.AuthenticationHeaderValue.Password = "admin"
folderApi.AuthenticationHeaderValue.Domain = "Domain" '''Only needed if AD/LDAP is being used
```

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Folder : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Folder
Inherits EktronBaseService
```

File

Folder.cs

Folder Methods

The methods of the Folder class are listed here.

Public Methods

	Name	Description
	AddFolder (see page 1237)	Adds a folder. This method can be accessed through the Folder.asmx file located in:
	CreateFolder (see page 1238)	Creates a folder. This method can be accessed through the Folder.asmx file located in:
	DeleteFolder (see page 1238)	Deletes folder by folder ID. This method can be accessed through the Folder.asmx file located in:
	GetChildFolders (see page 1238)	This is the overview for the GetChildFolders method overload.
	GetFolder (see page 1240)	This is the overview for the GetFolder method overload.
	GetFolderID (see page 1241)	This is the overview for the GetFolderID method overload.
	GetPath (see page 1242)	Finds a parent ID by using a given folder ID. This method can be accessed through the Folder.asmx file located in:
	RenameFolder (see page 1243)	Renames folder to specified title. This method can be accessed through the Folder.asmx file located in:
	UpdateFolder (see page 1243)	This is the overview for the UpdateFolder method overload.

Folder.AddFolder Method

Adds a folder.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddFolder", Description = "Adds a folder")]
public void AddFolder(ref FolderRequest FolderReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "AddFolder", Description = "Adds a folder")>
Public Function AddFolder(ByRef FolderReq As FolderRequest) As void
```

Parameters

Parameters	Description
ref FolderRequest FolderReq	Folder (see page 1236) Request Object.

Folder.CreateFolder Method

Creates a folder.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CreateFolder", Description = "Creates a folder with the given path")]
public bool CreateFolder(string FolderPath);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CreateFolder", Description = "Creates a folder with the given path")>
Public Function CreateFolder(FolderPath As string) As bool
```

Parameters

Parameters	Description
string FolderPath	The path on the IIS server. (Required)

Folder.DeleteFolder Method

Deletes folder by folder ID.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteFolder", Description = "Deletes folder specified by folder
id")]
public bool DeleteFolder(long FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteFolder", Description = "Deletes folder specified by folder
id")>
Public Function DeleteFolder(FolderID As long) As bool
```

Parameters

Parameters	Description
long FolderID	The ID of the folder to be deleted.

GetChildFolders Method

This is the overview for the GetChildFolders method overload.

Overload List

	Name	Description
	Folder.GetChildFolders (long, bool, Ektron.Cms.Common.EkEnumeration.FolderOrderBy) (see page 1239)	Loads all the subfolders for a given folder ID. This method can be accessed through the Folder.asmx file located in:
	Folder.GetChildFolders (long, bool, string) (see page 1239)	Loads all the subfolders for a given folder ID. This method can be accessed through the Folder.asmx file located in:

Folder.GetChildFolders Method (long, bool, Ektron.Cms.Common.EkEnumeration.FolderOrderBy)

Loads all the subfolders for a given folder ID.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetChildFolders", Description = "Get all the subfolders for a given
folder ID")]
public FolderData[] GetChildFolders(long FolderID, bool Recursive,
Ektron.Cms.Common.EkEnumeration.FolderOrderBy OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetChildFolders", Description = "Get all the subfolders for a given
folder ID")>
Public Function GetChildFolders(FolderID As long, Recursive As bool, OrderBy As
Ektron.Cms.Common.EkEnumeration.FolderOrderBy) As FolderData()
```

Parameters

Parameters	Description
long FolderID	The folder ID containing the subfolders.
bool Recursive	True = recursive look up. False = none
Ektron.Cms.Common.EkEnumeration.FolderOrderBy OrderBy	An order by clause.

Folder.GetChildFolders Method (long, bool, string)

Loads all the subfolders for a given folder ID.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetChildFolders_Obsolete", Description = "This method is obsolete.
Use new GetChildFolders method with FolderOrderBy enum parameter")]
[Obsolete("Call New GetChildFolders with FolderOrderBy enum parameter")]
public FolderData[] GetChildFolders(long FolderID, bool Recursive, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetChildFolders_Obsolete", Description = "This method is obsolete.
Use new GetChildFolders method with FolderOrderBy enum parameter")>
<Obsolete("Call New GetChildFolders with FolderOrderBy enum parameter")>
Public Function GetChildFolders(FolderID As long, Recursive As bool, OrderBy As string) As
FolderData()
```

Parameters

Parameters	Description
long FolderID	The folder ID containing the subfolders.
bool Recursive	True = recursive look up. False = none
string OrderBy	An order by clause.

GetFolder Method

This is the overview for the GetFolder method overload.

Description

This method is overloaded.

- GetFolder (Long) - retrieves the folder's details by providing the folder ID.
- GetFolder (Long, Bool, Bool) - retrieves the folder's details by providing the folder ID. In addition, this method allows you retrieve taxonomy and flag information for a folder.

Overload List

	Name	Description
	Folder.GetFolder (long) (see page 1240)	Retrieves the folder's details. This method can be accessed through the Folder.asmx file located in:
	Folder.GetFolder (long, bool, bool) (see page 1241)	Retrieves the folder's details. This method can be accessed through the Folder.asmx file located in:

Folder.GetFolder Method (long)

Retrieves the folder's details.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolder", Description = "Retrieves the folder's details")]
public FolderData GetFolder(long FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolder", Description = "Retrieves the folder's details")>
Public Function GetFolder(FolderID As long) As FolderData
```

Parameters

Parameters	Description
long FolderID	The folder's ID.

Folder.GetFolder Method (long, bool, bool)

Retrieves the folder's details.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolderByParam", Description = "Retrieves the folder's details")]
public FolderData GetFolder(long FolderID, bool getTaxonomy, bool getFlags);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolderByParam", Description = "Retrieves the folder's details")>
Public Function GetFolder(FolderID As long, getTaxonomy As bool, getFlags As bool) As
FolderData
```

Parameters

Parameters	Description
long FolderID	The folder's ID.
bool getTaxonomy	True if folder taxonomy required.
bool getFlags	True if folder flag required.

GetFolderID Method

This is the overview for the GetFolderID method overload.

Description

This method is overloaded.

- GetFolderID (Long) - finds the folder ID by providing a content item's ID.
- GetFolderID (String, Long) - finds the folder ID by providing the folder's name and in the parent folder's ID.

Overload List

	Name	Description
	Folder.GetFolderID (long) (see page 1241)	Finds the folder ID by using the content's ID. This method can be accessed through the Folder.asmx file located in:
	Folder.GetFolderID (string, long) (see page 1242)	Finds the folder ID by folder name and in a parent folder specified. This method can be accessed through the Folder.asmx file located in:

Folder.GetFolderID Method (long)

Finds the folder ID by using the content's ID.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
```

```
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolderID", Description = "Finds the folder ID by using the
content's ID")]
public long GetFolderID(long ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolderID", Description = "Finds the folder ID by using the
content's ID")>
Public Function GetFolderID(ContentID As long) As long
```

Parameters

Parameters	Description
long ContentID	The content's ID.

Folder.GetFolderID Method (string, long)

Finds the folder ID by folder name and in a parent folder specified.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolderIDByName", Description = "Finds the folder ID by folder
name and in a parent folder specified.")]
public long GetFolderID(string FolderName, long ParentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolderIDByName", Description = "Finds the folder ID by folder
name and in a parent folder specified.")>
Public Function GetFolderID(FolderName As string, ParentId As long) As long
```

Parameters

Parameters	Description
string FolderName	Name of the folder ID.
long ParentId	Look in a folder.

Returns

Returns the folder ID. Returns 0 if it does not exist.

Folder.GetPath Method

Finds a parent ID by using a given folder ID.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPath", Description = "Gets path to folder by folder ID")]
public string GetPath(long FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPath", Description = "Gets path to folder by folder ID")>
Public Function GetPath(FolderID As long) As string
```

Parameters

Parameters	Description
long FolderID	The folder's ID.

Folder.RenameFolder Method

Renames folder to specified title.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RenameFolder", Description = "Renames the folder by ID.")]
public bool RenameFolder(long FolderID, string Title);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RenameFolder", Description = "Renames the folder by ID.")>
Public Function RenameFolder(FolderID As long, Title As string) As bool
```

Parameters

Parameters	Description
long FolderID	The folder's ID.
string Title	The folder's new title

UpdateFolder Method

This is the overview for the UpdateFolder method overload.

Description

This method is overloaded.

- UpdateFolder (FolderData) (see page 1243)
- UpdateFolder (FolderRequest) (see page 1244)

Overload List

	Name	Description
	Folder.UpdateFolder (FolderData) (see page 1243)	Updates a folder. This method can be accessed through the Folder.asmx file located in:
	Folder.UpdateFolder (FolderRequest) (see page 1244)	Updates a folder. This method can be accessed through the Folder.asmx file located in:

Folder.UpdateFolder Method (FolderData)

Updates a folder.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateFolder[FolderData]", Description = "Updates a folder")]
public void UpdateFolder(FolderData FolderReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateFolder[FolderData]", Description = "Updates a folder")>
Public Function UpdateFolder(FolderReq As FolderData) As void
```

Parameters

Parameters	Description
FolderData FolderReq	Folder (see page 1236) Request Object.

Folder.UpdateFolder Method (FolderRequest)

Updates a folder.

This method can be accessed through the Folder.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Folder.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateFolder", Description = "Updates a folder")]
public void UpdateFolder(FolderRequest FolderReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateFolder", Description = "Updates a folder")>
Public Function UpdateFolder(FolderReq As FolderRequest) As void
```

Parameters

Parameters	Description
FolderRequest FolderReq	Folder (see page 1236) Request Object.

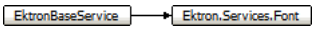
Font Class

This section describes the Web Service methods used with Fonts. The Font Web service is a wrapper for the Font API (see page 669): `Ektron.Cms.API.Font`. These methods can be accessed through the Font.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Font.asmx
```

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class Font : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class Font
Inherits EktronBaseService
```

File

Font.cs

Font Methods

The methods of the Font class are listed here.

Public Methods

	Name	Description
	AddFont (see page 1245)	Adds a new font to the system. This method can be accessed through the Font.asmx file located in:
	DeleteFont (see page 1245)	Deletes a font from the system. This method can be accessed through the Font.asmx file located in:
	GetAllFonts (see page 1246)	Loads all of the fonts as a array of FontData. This method can be accessed through the Font.asmx file located in:
	GetFont (see page 1246)	Loads the required font using the font's ID. This method can be accessed through the Font.asmx file located in:
	GetFontConfigList (see page 1247)	Returns all of the font details in a drop down control. This method can be accessed through the Font.asmx file located in:
	UpdateFont (see page 1247)	Updates the font. This method can be accessed through the Font.asmx file located in:

Font.AddFont Method

Adds a new font to the system.

This method can be accessed through the Font.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Font.asmx`

C#

```
public void AddFont(Collection data);
```

Visual Basic

```
Public Function AddFont(data As Collection) As void
```

Parameters

Parameters	Description
Collection data	Font (see page 1244) data as a value key pair.

Font.DeleteFont Method

Deletes a font from the system.

This method can be accessed through the Font.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Font.asmx`

C#

```
public void DeleteFont(Collection data);
```

Visual Basic

```
Public Function DeleteFont(data As Collection) As void
```

Parameters

Parameters	Description
Collection data	Required font data (for example, font ID).

Remarks

A user with administrator privileges must be logged in to perform this operation.

Font.GetAllFonts Method

Loads all of the fonts as a array of FontData.

This method can be accessed through the Font.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Font.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllFonts", Description = "")]
public FontData[] GetAllFonts();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllFonts", Description = "")>
Public Function GetAllFonts() As FontData()
```

Returns

Ektron.Cms.FontData()

Remarks

User (see page 1369) login is required to perform this operation.

Font.GetFont Method

Loads the required font using the font's ID.

This method can be accessed through the Font.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Font.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFont", Description = "")]
public FontData GetFont(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFont", Description = "")>
Public Function GetFont(Id As long) As FontData
```

Parameters

Parameters	Description
long Id	The font's numeric ID.

Returns

A numeric value.

Font.GetFontConfigList Method

Returns all of the font details in a drop down control.

This method can be accessed through the Font.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Font.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFontConfigList", Description = "")]
public string GetFontConfigList();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFontConfigList", Description = "")>
Public Function GetFontConfigList() As string
```

Returns

String

Font.UpdateFont Method

Updates the font.

This method can be accessed through the Font.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Font.asmx`

C#

```
public void UpdateFont(Collection data);
```

Visual Basic

```
Public Function UpdateFont(data As Collection) As void
```

Parameters

Parameters	Description
Collection data	The font's data.

Remarks

A user with administrator privileges must be logged in to perform this operation.

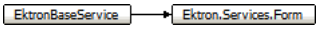
Form Class

This section describes the Web Service methods used with forms. The Form Web service is a wrapper for the Form API (see page 669): `Ektron.Cms.API.Content.Form` (see page 871). These methods can be accessed through the form.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Form.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Form : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Form
Inherits EktronBaseService
```

File

Form.cs

Form Methods

The methods of the Form class are listed here.

Public Methods

	Name	Description
	AssignFormTask (see page 1248)	Assigns a form submission task, as specified in FormInfo (see page 588), of the SubmittedDataXml. This method can be accessed through the Form.asmx file located in:
	EmailFormFieldData (see page 1249)	Sends an email, as specified in FormInfo (see page 588), of the SubmittedDataXml. This method can be accessed through the Form.asmx file located in:
	GetAllForms (see page 1249)	This is the overview for the GetAllForms method overload.
	GetForm (see page 1251)	Retrieves information about the specified form. This method can be accessed through the Form.asmx file located in:
	GetFormFieldData (see page 1251)	This is the overview for the GetFormFieldData method overload.
	GetFormFieldList (see page 1253)	Retrieves the list of form fields defined in the specified form. This method can be accessed through the Form.asmx file located in:
	GetFormFieldQuestions (see page 1253)	Retrieves the table of form fields and their questions in a specified form. In a CMS form, the form question is defined by the user in the descriptive name field in the form element dialog during design time. This method can be accessed through the Form.asmx file located in:
	GetFormTitle (see page 1254)	Same as GetFormById, but returns only the form's title. This method can be accessed through the Form.asmx file located in:
	SubmitFormFieldData (see page 1254)	Adds form data as if it was submitted during a browser post to the web server. This method can be accessed through the Form.asmx file located in:
	UpdateFormFieldData (see page 1255)	Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form. This method can be accessed through the Form.asmx file located in:

Form.AssignFormTask Method

Assigns a form submission task, as specified in FormInfo (see page 588), of the SubmittedDataXml.

This method can be accessed through the Form.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Form.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignFormTask", Description = "")]
public bool AssignFormTask(FormData FormInfo, string SubmittedDataXml);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignFormTask", Description = "")>
```

```
Public Function AssignFormTask(FormInfo As FormData, SubmittedDataXml As string) As bool
```

Parameters

Parameters	Description
FormData FormInfo	The properties of the form associated with the data. This information may be retrieved by calling GetFormById. Value Type: Cms.FormData (Ektron.Cms.FormData)
string SubmittedDataXml	The form data submitted in XML format. Value Type: String (System.String)

Returns

Returns True if the task was successfully assigned. Boolean (System.Boolean)

Remarks

This routine is called automatically when calling SubmitFormFieldData (see page 1254) if SendNotification is True and the form is configured to assign a task when data is submitted. The SubmittedDataXml is typically retrieved by calling SubmitFormFieldData (see page 1254).

Form.EmailFormFieldData Method

Sends an email, as specified in FormInfo (see page 588), of the SubmittedDataXml.

This method can be accessed through the Form.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/Form.asmx`

C#

```
public void EmailFormFieldData(FormData FormInfo, string SubmittedDataXml);
```

Visual Basic

```
Public Function EmailFormFieldData(FormInfo As FormData, SubmittedDataXml As string) As void
```

Parameters

Parameters	Description
FormData FormInfo	The properties of the form associated with the data. This information may be retrieved by calling GetFormById. Value Type: Cms.FormData (Ektron.Cms.FormData)
string SubmittedDataXml	The form data submitted in XML format. Value Type: String (System.String)

Remarks

This routine is called automatically when calling SubmitFormFieldData (see page 1254) if SendNotification is True and the form is configured to send an email notification when data is submitted. The SubmittedDataXml is typically retrieved by calling SubmitFormFieldData (see page 1254).

GetAllForms Method

This is the overview for the GetAllForms method overload.

Description

This method is overloaded.

- GetAllForms () - returns all of the forms as an array of FormData.
- GetAllForms (Long, String, Int ContentAPI.FormResultType) - retrieves an array of all the forms contained with the specified folder. Forms may be limited to a single language and type.

Overload List

	Name	Description
	Form.GetAllForms () (see page 1250)	Returns all of the forms as an array of FormData. This method can be accessed through the Form.asmx file located in:
	Form.GetAllForms (long, string, int, ContentAPI.FormResultType) (see page 1250)	Retrieves an array of all the forms contained with the specified folder. Forms may be limited to a single language and type. This method can be accessed through the Form.asmx file located in:

Form.GetAllForms Method ()

Returns all of the forms as an array of FormData.

This method can be accessed through the Form.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Form.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllForms", Description = "")]
public FormData[] GetAllForms();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllForms", Description = "")>
Public Function GetAllForms() As FormData()
```

Returns

Ektron.Cms.FormData()

Remarks

User (see page 1369) login required to perform this operation.

Form.GetAllForms Method (long, string, int, ContentAPI.FormResultType)

Retrieves an array of all the forms contained with the specified folder. Forms may be limited to a single language and type.

This method can be accessed through the Form.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Form.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllForms By Id, OrderBy, LanguageID and Type", Description = "")]
public FormData[] GetAllForms(long Id, string OrderBy, int LanguageId,
ContentAPI.FormResultType Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllForms By Id, OrderBy, LanguageID and Type", Description = "")>
Public Function GetAllForms(Id As long, OrderBy As string, LanguageId As Integer, Type As
ContentAPI.FormResultType) As FormData()
```

Parameters

Parameters	Description
long Id	Specifies the folder.
string OrderBy	Specifies how to sort the forms
int LanguageId	Specifies a language or EkConstants.g_AllContentLanguages (-1) for all languages.
ContentAPI.FormResultType Type	Specifies whether only active forms or archived forms should be returned.

Remarks

Requires authentication.

Form.GetForm Method

Retrieves information about the specified form.

This method can be accessed through the Form.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Form.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetForm", Description = "")]
public FormData GetForm(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetForm", Description = "")>
Public Function GetForm(Id As long) As FormData
```

Parameters

Parameters	Description
long Id	Specifies the form.

Returns

Returns information about the form.

Remarks

Does not require authentication.

GetFormFieldData Method

This is the overview for the GetFormFieldData method overload.

Description

This method is overloaded.

- GetFormFieldData (Long, Long) - retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.
- GetFormFieldData (Long, String, String) - retrieves the submitted data for all fields from a specified form. All the data may be retrieved or just data submitted within a given date range.

Overload List

	Name	Description
	Form.GetFormFieldData (long, long) (see page 1252)	Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form. This method can be accessed through the Form.asmx file located in:
	Form.GetFormFieldData (long, string, string) (see page 1252)	Retrieves the submitted data for all fields from a specified form. All the data may be retrieved or just data submitted within a given date range. This method can be accessed through the Form.asmx file located in:

Form.GetFormFieldData Method (long, long)

Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

This method can be accessed through the Form.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Form.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFormFieldData By FormID and FormDataId", Description = "")]
public FormSubmittedData GetFormFieldData(long FormId, long FormDataId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFormFieldData By FormID and FormDataId", Description = "")>
Public Function GetFormFieldData(FormId As long, FormDataId As long) As FormSubmittedData
```

Parameters

Parameters	Description
long FormId	Specifies the form. Value Type: Int32 (System.Int32)
long FormDataId	Specifies the instance of submitted data. Value Type: Int32 (System.Int32)

Returns

Returns the submitted data or null/Nothing, if none found. Cms.FormSubmittedData (Ektron.Cms.FormSubmittedData)

Remarks

Does not require authentication.

Form.GetFormFieldData Method (long, string, string)

Retrieves the submitted data for all fields from a specified form. All the data may be retrieved or just data submitted within a given date range.

This method can be accessed through the Form.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Form.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFormFieldData By FormId, StartDate and EndDate", Description =
"")]
public FormSubmittedData[] GetFormFieldData(long FormId, string StartDate, string EndDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFormFieldData By FormId, StartDate and EndDate", Description =
"")>
Public Function GetFormFieldData(FormId As long, StartDate As string, EndDate As string) As
FormSubmittedData()
```

Parameters

Parameters	Description
long FormId	Specifies the form. Value Type: Int32 (System.Int32)
string StartDate	Optionally limit the data returned to include only those submitted on or after the given date. Value Type: String (System.String)
string EndDate	Optionally limit the data returned to include only those submitted on or before the given date. Value Type: String (System.String)

Returns

Returns an array of submitted data. Cms.FormSubmittedData (Ektron.Cms.FormSubmittedData)

Remarks

Does not require authentication.

Form.GetFormFieldList Method

Retrieves the list of form fields defined in the specified form.

This method can be accessed through the Form.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Form.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFormFieldList", Description = "")]
public FormFieldList GetFormFieldList(long FormId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFormFieldList", Description = "")>
Public Function GetFormFieldList(FormId As long) As FormFieldList
```

Parameters

Parameters	Description
long FormId	Specifies the form. Value Type: Int32 (System.Int32)

Returns

The list for defined fields in the given form. Cms.FormFieldList (see page 1636) (Ektron.Cms.FormFieldList (see page 1636))

Form.GetFormFieldQuestions Method

Retrieves the table of form fields and their questions in a specified form. In a CMS form, the form question is defined by the user in the descriptive name field in the form element dialog during design time.

This method can be accessed through the Form.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Form.asmx
```

C#

```
public Hashtable GetFormFieldQuestions(long FormId);
```

Visual Basic

```
Public Function GetFormFieldQuestions(FormId As long) As Hashtable
```

Parameters

Parameters	Description
long FormId	Specifies the form. Value Type: Int32 (System.Int32)

Returns

a table of defined fields with the corresponding questions in a given form.

Description

[WebMethod(MessageName="GetFormFieldQuestions",Description="")]

Remarks

these questions is mainly used for reporting.

Form.GetFormTitle Method

Same as GetFormById, but returns only the form's title.

This method can be accessed through the Form.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Form.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFormTitle", Description = "")]
public string GetFormTitle(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFormTitle", Description = "")>
Public Function GetFormTitle(Id As long) As string
```

Parameters

Parameters	Description
long Id	Specifies the form.

Returns

The title of the specified form.

Remarks

Does not require authentication.

Form.SubmitFormFieldData Method

Adds form data as if it was submitted during a browser post to the web server.

This method can be accessed through the Form.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Form.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SubmitFormFieldData", Description = "")]
public void SubmitFormFieldData(long FormId, FormSubmittedData Data, bool SendNotification,
string SubmittedDataXml);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SubmitFormFieldData", Description = "")>
Public Function SubmitFormFieldData(FormId As long, Data As FormSubmittedData,
SendNotification As bool, SubmittedDataXml As string) As void
```

Parameters

Parameters	Description
long FormId	Specifies the form that corresponds to the data. Value Type: Int32 (System.Int32)
FormSubmittedData Data	The data to submit. Value Type: Cms.FormSubmittedData (Ektron.Cms.FormSubmittedData)
bool SendNotification	(Optional) Specifies whether to send notifications that a form submission has occurred or not. For example, whether to send an email or assign a form submission task as configured for the form. Value Type: Boolean (System.Boolean)
string SubmittedDataXml	(Optional) XML packet of submitted data. Reference Type: String (System.String)

Form.UpdateFormFieldData Method

Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

This method can be accessed through the Form.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/Form.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateFormFieldData", Description = "")]
public void UpdateFormFieldData(long FormId, long FormDataId, FormSubmittedData SubmittedData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateFormFieldData", Description = "")>
Public Function UpdateFormFieldData(FormId As long, FormDataId As long, SubmittedData As
FormSubmittedData) As void
```

Parameters

Parameters	Description
long FormId	Specifies the form. Value Type: Int32 (System.Int32)
long FormDataId	Specifies the instance of submitted data. Value Type: Int32 (System.Int32)
FormSubmittedData SubmittedData	The field data values to be updated. Value Type: Cms.FormSubmittedData (Ektron.Cms.FormSubmittedData)

Remarks

Does not require authentication.

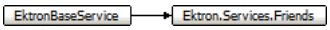
Friends Class

This section describes the Web Service methods used with Ektron's Friends feature. This feature is a part Ektron's Community (see page 750) Platform. The Friends Web service is a wrapper for the Friends API (see page 669): `Ektron.Cms.API.Community.Friends` (see page 777). These methods can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
public class Friends : EktronBaseService;
```

Visual Basic

```
Public Class Friends
Inherits EktronBaseService
```

File

Friends.cs

Friends Methods

The methods of the Friends class are listed here.

Public Methods

	Name	Description
	AcceptPendingFriend (see page 1257)	Accepts a friend request from one user to another. This method can be accessed through the Friends.asmx file located in:
	AddFriend (see page 1257)	This is the overview for the AddFriend method overload.
	AddPendingFriend (see page 1258)	Adds a pending friend. Sends the other user a friend invitation. This method can be accessed through the Friends.asmx file located in:
	AddSelectedFriend (see page 1259)	Adds a friend to a user's selected friends list. This method can be accessed through the Friends.asmx file located in:
	DeleteFriend (see page 1259)	Deletes a friend from a user's friends list. This method can be accessed through the Friends.asmx file located in:
	DeletePendingFriendRequest (see page 1260)	This is the overview for the DeletePendingFriendRequest method overload.
	DeleteSelectedFriend (see page 1261)	Delete's a friend from a user's selected friends list. This method can be accessed through the Friends.asmx file located in:
	GetFriends (see page 1261)	This is the overview for the GetFriends method overload.
	GetFriendsDirectory (see page 1264)	Retrieves friends directory. This method can be accessed through the Friends.asmx file located in:
	GetFriendStatus (see page 1264)	Get another user's friend status in relation to you. For instance, user "A" may be a friend to User (see page 1369) "B". User (see page 1369) "C" may be a pending friend to User (see page 1369) "D". This method can be accessed through the Friends.asmx file located in:
	GetInvitedUsers (see page 1265)	Gets a list of sent friend invitations for a given user. This method can be accessed through the Friends.asmx file located in:
	GetPendingFriends (see page 1266)	Gets a list of users that have sent another user friend invitations. This only returns users who's invitations have not yet been accepted and are still pending. This method can be accessed through the Friends.asmx file located in:
	GetSelectedFriends (see page 1266)	Get a user's selected friends list. This method can be accessed through the Friends.asmx file located in:
	GetSentFriendInvitations (see page 1267)	Gets a list of pending Friend invitations sent out by a user. This method can be accessed through the Friends.asmx file located in:
	InviteUsers (see page 1267)	Send a friend invitation a list of users and\or email addresses. This method can be accessed through the Friends.asmx file located in:
	IsFriend (see page 1268)	Returns true if FriendUserId and UserID are friends. This method can be accessed through the Friends.asmx file located in:
	IsSelectedFriend (see page 1268)	Check if a user is a selected friend of another user. This method can be accessed through the Friends.asmx file located in:

	MoveFriend (see page 1269)	Moves a friend into another friend directory. This method can be accessed through the Friends.asmx file located in:

Friends.AcceptPendingFriend Method

Accepts a friend request from one user to another.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AcceptPendingFriend", Description = "Accepts a friend request from
one user to another.")]
public void AcceptPendingFriend(long fromUserId, long toUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AcceptPendingFriend", Description = "Accepts a friend request from
one user to another.")>
Public Function AcceptPendingFriend(fromUserId As long, toUserId As long) As void
```

Parameters

Parameters	Description
long fromUserId	ID of user who sent friend request.
long toUserId	ID of user who received friend request and is accepting it.

AddFriend Method

This is the overview for the AddFriend method overload.

Description

This method is overloaded.

- AddFriend (Long, Long) - adds a user to another user's friends list based on the ID of the user who is being added as a friend and the ID of the user adding the friend.
- AddFriend (Long, Long, Long) - adds a user to a specified directory of another user's friends list based on the ID of the user who is being added as a friend, the ID of the user adding the friend and the ID of the directory where the user is being added.

Overload List

	Name	Description
	Friends.AddFriend (long, long) (see page 1257)	Adds a user to another user's friends list. This method can be accessed through the Friends.asmx file located in:
	Friends.AddFriend (long, long, long) (see page 1258)	Adds a user to another user's friends list in a specific sub directory. This method can be accessed through the Friends.asmx file located in:

Friends.AddFriend Method (long, long)

Adds a user to another user's friends list.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddFriend", Description = "Adds a user to another user's friends
list.")]
public void AddFriend(long friendUserId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddFriend", Description = "Adds a user to another user's friends
list.")>
Public Function AddFriend(friendUserId As long, userId As long) As void
```

Parameters

Parameters	Description
long friendUserId	The user ID to add to users friends list.
long userId	The ID of the user adding friend.

Friends.AddFriend Method (long, long, long)

Adds a user to another user's friends list in a specific sub directory.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddFriend for directory", Description = "Adds a user to aother
user's friends list in a specific sub directory.")]
public void AddFriend(long friendUserId, long userId, long directoryId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddFriend for directory", Description = "Adds a user to aother
user's friends list in a specific sub directory.")>
Public Function AddFriend(friendUserId As long, userId As long, directoryId As long) As void
```

Parameters

Parameters	Description
long friendUserId	The user ID of friend to add to users friends list.
long userId	The ID of user adding friend.
long directoryId	The ID of directory to place friend into.

Friends.AddPendingFriend Method

Adds a pending friend. Sends the other user a friend invitation.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddPendingFriend", Description = "Adds a pending friend. Sends the
other user a friend invitation.")]
```

```
public void AddPendingFriend(long friendUserId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddPendingFriend", Description = "Adds a pending friend. Sends the
other user a friend invitation.")>
Public Function AddPendingFriend(friendUserId As long, userId As long) As void
```

Parameters

Parameters	Description
long friendUserId	The user receiving a friend request.
long userId	The user sending a friend request.

Friends.AddSelectedFriend Method

Adds a friend to a user's selected friends list.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddSelectedFriend", Description = "Adds a friend to a user's
selected friends list.")]
public void AddSelectedFriend(long selectedFriendUserId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddSelectedFriend", Description = "Adds a friend to a user's
selected friends list.")>
Public Function AddSelectedFriend(selectedFriendUserId As long, userId As long) As void
```

Parameters

Parameters	Description
long selectedFriendUserId	The ID of the friend to be added to the selected list.
long userId	The ID of the user adding the selected friend.

Friends.DeleteFriend Method

Deletes a friend from a user's friends list.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteFriend", Description = "Deletes a friend from a user's friends
list.")]
public void DeleteFriend(long friendUserId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteFriend", Description = "Deletes a friend from a user's friends
list.")>
Public Function DeleteFriend(friendUserId As long, userId As long) As void
```

Parameters

Parameters	Description
long friendUserId	User (see page 1369) ID of the friend to delete from a user's friends list.
long userId	The ID of a user deleting the friend.

DeletePendingFriendRequest Method

This is the overview for the DeletePendingFriendRequest method overload.

Description

This method is overloaded.

- DeletePendingFriendRequest (Long, Long) - deletes a pending friend request by providing the ID of the user who sent the original friend request and the ID of the user who received the request.
- DeletePendingFriendRequest (Long, String) - Ektron.Services.Friends.DeletePendingFriendRequest@int@stringdeletes a pending friend request by providing the ID of the user who sent the original friend request and the email address of the user who received the request.

Overload List

	Name	Description
	Friends.DeletePendingFriendRequest (long, long) (see page 1260)	Deletes a pending friend request. This method can be accessed through the Friends.asmx file located in:
	Friends.DeletePendingFriendRequest (long, string) (see page 1260)	Deletes a pending friend request. This method can be accessed through the Friends.asmx file located in:

Friends.DeletePendingFriendRequest Method (long, long)

Deletes a pending friend request.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeletePendingFriendRequest", Description = "Deletes a pending friend
request.")]
public void DeletePendingFriendRequest(long fromUserId, long toUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeletePendingFriendRequest", Description = "Deletes a pending friend
request.")>
Public Function DeletePendingFriendRequest(fromUserId As long, toUserId As long) As void
```

Parameters

Parameters	Description
long fromUserId	The ID of the user who sent original friend request.
long toUserId	The ID of the user who received friend request.

Friends.DeletePendingFriendRequest Method (long, string)

Deletes a pending friend request.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeletePendingFriendRequest by email address", Description = "Deletes
a pending friend request.")]
public void DeletePendingFriendRequest(long fromUserId, string toEmailAddress);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeletePendingFriendRequest by email address", Description = "Deletes
a pending friend request.")>
Public Function DeletePendingFriendRequest(fromUserId As long, toEmailAddress As string) As
void
```

Parameters

Parameters	Description
long fromUserId	The ID of the user who sent original friend request.
string toEmailAddress	The Email address of the user who received friend request.

Friends.DeleteSelectedFriend Method

Delete's a friend from a user's selected friends list.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteSelectedFriend", Description = "Delete's a friend from a
user's selected friends list.")]
public void DeleteSelectedFriend(long selectedFriendUserId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteSelectedFriend", Description = "Delete's a friend from a
user's selected friends list.")>
Public Function DeleteSelectedFriend(selectedFriendUserId As long, userId As long) As void
```

Parameters

Parameters	Description
long selectedFriendUserId	User (see page 1369) ID of the friend to delete from user's selected friends list.
long userId	ID of the user who is deleting the friend.

GetFriends Method

This is the overview for the GetFriends method overload.

Description

This method is overloaded.

- GetFriends (Long) - returns a list of friends for a user by providing the user's ID.
- GetFriends (Long, PagingInfo) - returns a list of friends for a user by providing the user's ID. This method allows for paging of the list.

- GetFriends (Long, PagingInfo, Long) - returns a user's list of friends in a specific subdirectory by providing the user's ID and the ID of the sub directory. This method allows for paging of the list.
- GetFriends (Long, Long) - returns a user's list of friends in a specific subdirectory by providing the user's ID and the ID of the sub directory.

Overload List

	Name	Description
	Friends.GetFriends (long) (see page 1262)	Returns a list of friends for a user. This method can be accessed through the Friends.asmx file located in:
	Friends.GetFriends (long, long) (see page 1263)	Returns a user's list of friends in a specific subdirectory. This method can be accessed through the Friends.asmx file located in:
	Friends.GetFriends (long, PagingInfo) (see page 1262)	Returns a list of friends for a user. This method can be accessed through the Friends.asmx file located in:
	Friends.GetFriends (long, PagingInfo, long) (see page 1263)	Returns a user's list of friends in a specific subdirectory. This method can be accessed through the Friends.asmx file located in:

Friends.GetFriends Method (long)

Returns a list of friends for a user.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFriends", Description = "Returns a list of friends for a user.")]
public DirectoryUserData[] GetFriends(long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFriends", Description = "Returns a list of friends for a user.")>
Public Function GetFriends(userId As long) As DirectoryUserData()
```

Parameters

Parameters	Description
long userId	ID of user to retrieve friends for.

Returns

An array of DirectoryUserData.

Friends.GetFriends Method (long, PagingInfo)

Returns a list of friends for a user.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFriends with paging info", Description = "Returns a list of
friends for a user.")]
public DirectoryUserData[] GetFriends(long userId, PagingInfo pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFriends with paging info", Description = "Returns a list of
friends for a user.")>
Public Function GetFriends(userId As long, pagingInfo As PagingInfo) As DirectoryUserData()
```

Parameters

Parameters	Description
long userId	ID of user to retrieve friends for.
PagingInfo pagingInfo	Supply paging information to control retrieval.

Returns

An array of DirectoryUserData.

Friends.GetFriends Method (long, PagingInfo, long)

Returns a user's list of friends in a specific subdirectory.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFriends with paging info for directory", Description = "Returns a
user's list of friends in a specific subdirectory.")]
public DirectoryUserData[] GetFriends(long userId, PagingInfo pagingInfo, long directoryID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFriends with paging info for directory", Description = "Returns a
user's list of friends in a specific subdirectory.")>
Public Function GetFriends(userId As long, pagingInfo As PagingInfo, directoryID As long) As
DirectoryUserData()
```

Parameters

Parameters	Description
long userId	The ID of a user to retrieve friends for.
PagingInfo pagingInfo	Supply paging information to control retrieval.
long directoryID	The ID of directory to retrieve friends from.

Returns

An array of DirectoryUserData.

Friends.GetFriends Method (long, long)

Returns a user's list of friends in a specific subdirectory.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFriends for directory", Description = "Returns a user's list of
friends in a specific subdirectory.")]
public DirectoryUserData[] GetFriends(long userId, long directoryID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFriends for directory", Description = "Returns a user's list of
friends in a specific subdirectory.")>
Public Function GetFriends(userId As long, directoryID As long) As DirectoryUserData()
```

Parameters

Parameters	Description
long userId	ID of a user to retrieve friends for.
long directoryID	ID of directory to retrieve friends from.

Returns

An array of DirectoryUserData.

Friends.GetFriendsDirectory Method

Retrieves friends directory.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFriendsDirectory", Description = "Retrieves friends directory.")]
public DirectoryAdvancedUserData GetFriendsDirectory(long userId, long directoryId, PagingInfo
pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFriendsDirectory", Description = "Retrieves friends directory.")>
Public Function GetFriendsDirectory(userId As long, directoryId As long, pagingInfo As
PagingInfo) As DirectoryAdvancedUserData
```

Parameters

Parameters	Description
long userId	The ID of the user who is retrieving a friends directory.
long directoryId	The ID of friends directory to retrieve.
PagingInfo pagingInfo	Supply paging information to control retrieval.

Returns

DirectoryAdvancedUserData

Friends.GetFriendStatus Method

Get another user's friend status in relation to you. For instance, user "A" may be a friend to User (see page 1369) "B". User (see page 1369) "C" may be a pending friend to User (see page 1369) "D".

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFriendStatus", Description = "Get another user's friend status in
relation to you.  For instance, user A may be  a friend to User B.  User C may be a pending
friend to User D.")]
public EkEnumeration.FriendStatus GetFriendStatus(long friendUserId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFriendStatus", Description = "Get another user's friend status in
relation to you.  For instance, user A may be  a friend to User B.  User C may be a pending
friend to User D.")>
Public Function GetFriendStatus(friendUserId As long, userId As long) As
EkEnumeration.FriendStatus
```

Parameters

Parameters	Description
long friendUserId	User (see page 1369) to get status.
long userId	User (see page 1369) seeking status.

Returns

FriendStatus enumeration.

Friends.GetInvitedUsers Method

Gets a list of sent friend invitations for a given user.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetInvitedUsers", Description = "Gets a list of sent friend
invitations for a given user.")]
public DirectoryUserData[] GetInvitedUsers(long userId, PagingInfo pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetInvitedUsers", Description = "Gets a list of sent friend
invitations for a given user.")>
Public Function GetInvitedUsers(userId As long, pagingInfo As PagingInfo) As
DirectoryUserData()
```

Parameters

Parameters	Description
long userId	ID of user who sent friend invitations.
PagingInfo pagingInfo	supply paging information to control retrieval.

Returns

An array of DirectoryUserData

Friends.GetPendingFriends Method

Gets a list of users that have sent another user friend invitations. This only returns users who's invitations have not yet been accepted and are still pending.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPendingFriends", Description = "Gets a list of users that have
sent another user friend invitations. This only returns users who's invitations have not yet
been accepted and are still pending.")]
public DirectoryUserData[] GetPendingFriends(long userId, PagingInfo pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPendingFriends", Description = "Gets a list of users that have
sent another user friend invitations. This only returns users who's invitations have not yet
been accepted and are still pending.")>
Public Function GetPendingFriends(userId As long, pagingInfo As PagingInfo) As
DirectoryUserData()
```

Parameters

Parameters	Description
long userId	ID of user who has received invitations.
PagingInfo pagingInfo	Supply paging information to control retrieval.

Returns

An array of DirectoryUserData.

Friends.GetSelectedFriends Method

Get a user's selected friends list.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetSelectedFriends", Description = "Get a list of user's selected
friends.")]
public DirectoryUserData[] GetSelectedFriends(long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetSelectedFriends", Description = "Get a list of user's selected
friends.")>
Public Function GetSelectedFriends(userId As long) As DirectoryUserData()
```

Parameters

Parameters	Description
long userId	ID of user to retrieve selected friends for.

Returns

An array of DirectoryUserData

Friends.GetSentFriendInvitations Method

Gets a list of pending Friend invitations sent out by a user.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetSentFriendInvitations", Description = "Gets a list of pending
Friend invitations sent out by a user.")]
public FriendInvitation[] GetSentFriendInvitations(long userId, PagingInfo pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetSentFriendInvitations", Description = "Gets a list of pending
Friend invitations sent out by a user.")>
Public Function GetSentFriendInvitations(userId As long, pagingInfo As PagingInfo) As
FriendInvitation()
```

Parameters

Parameters	Description
long userId	The ID of user who sent the invitations.
PagingInfo pagingInfo	Supply paging information to control retrieval.

Returns

An array of FriendInvitations.

Friends.InviteUsers Method

Send a friend invitation a list of users and\or email addresses.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "InviteUsers", Description = "Send a friend invitation a list of
users and\\or email addresses.")]
public Collection<Invitation> InviteUsers(InvitationSendRequestData invitationData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "InviteUsers", Description = "Send a friend invitation a list of
users and\\or email addresses.")>
Public Function InviteUsers(invitationData As InvitationSendRequestData) As
Collection<Invitation>
```

Parameters

Parameters	Description
InvitationSendRequestData invitationData	The invitation request data detailing invitation information.

Returns

A collection of invitation objects that have been sent out.

Friends.IsFriend Method

Returns true if FriendUserId and UserID are friends.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsFriend", Description = "Returns true if the FriendUserId and
UserID are friends.")]
public bool IsFriend(long friendUserId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsFriend", Description = "Returns true if the FriendUserId and
UserID are friends.")>
Public Function IsFriend(friendUserId As long, userId As long) As bool
```

Parameters

Parameters	Description
long friendUserId	First user ID.
long userId	Second user ID.

Returns

Returns true if FriendUserId and UserID are friends, else false.

Friends.IsSelectedFriend Method

Check if a user is a selected friend of another user.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsSelectedFriend", Description = "Check if a user is a selected
friend of another user.")]
public bool IsSelectedFriend(long friendUserId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsSelectedFriend", Description = "Check if a user is a selected
friend of another user.")>
Public Function IsSelectedFriend(friendUserId As long, userId As long) As bool
```

Parameters

Parameters	Description
long friendUserId	User (see page 1369) ID to check for in selected friends list.
long userId	ID of user who is checking for selected friend.

Returns

True if friendUserId is in the selected friends list of UserId, else false.

Friends.MoveFriend Method

Moves a friend into another friend directory.

This method can be accessed through the Friends.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Friends.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MoveFriend", Description = "Moves a friend into another friend
directory.")]
public void MoveFriend(long friendUserId, long userId, long directoryId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MoveFriend", Description = "Moves a friend into another friend
directory.")>
Public Function MoveFriend(friendUserId As long, userId As long, directoryId As long) As void
```

Parameters

Parameters	Description
long friendUserId	ID of friend to move.
long userId	ID of user moving the friend.
long directoryId	ID of the directory to move the friend into.

Library Class

This section describes the Web Service methods used with the Library. The Library Web service is a wrapper for the Library API (see page 669): `Ektron.Cms.API.Library` (see page 695). These methods can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

Web Services (see page 1083) examples involving content are available in the Developer sample site. You can download this site from:

http://www.ektron.com/cms400-web-cms.aspx?id=5278

Once the developer site is installed, example code files are located in your site root. For example:

```
C:\Inetpub\wwwroot\CMS400Developer\Developer\dotNetAssemblies\WebServicesAPI\LibraryAPI
```

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header. For example:

```
wslibraryApi.AuthenticationHeaderValue = New localhost.AuthenticationHeader()
wslibraryApi.AuthenticationHeaderValue.Username = "admin"
wslibraryApi.AuthenticationHeaderValue.Password = "admin"
wslibraryApi.AuthenticationHeaderValue.Domain = "Domain" '''Only needed if AD/LDAP is being
used
```

Class Hierarchy

C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Library : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Library
Inherits EktronBaseService
```

File

Library.cs

Library Methods

The methods of the Library class are listed here.

Public Methods

	Name	Description
	AddLoadBalanceItem (see page 1270)	Adds a load balance path for files and images. This method can be accessed through the Library.asmx file located in:
	DeleteLibraryItem (see page 1271)	Deletes the library item by extension. This method can be accessed through the Library.asmx file located in:
	DeleteLibraryItemById (see page 1271)	Deletes the library items from the system. This method can be accessed through the Library.asmx file located in:
	DeleteLoadBalanceItem (see page 1272)	Deletes the load balance item from the system. This method can be accessed through the Library.asmx file located in:
	GetAllChildLibItems (see page 1272)	Loads all of the library items by type. This method can be accessed through the Library.asmx file located in:
	GetAllLoadBalancePaths (see page 1273)	This is the overview for the GetAllLoadBalancePaths method overload.
	GetLibraryItem (see page 1274)	This is the overview for the GetLibraryItem method overload.
	GetLibrarySettings (see page 1275)	Loads the library settings. This method can be accessed through the Library.asmx file located in:
	GetLibraryTypes (see page 1276)	Loads the library types. This method can be accessed through the Library.asmx file located in:
	GetLoadBalancePath (see page 1276)	Gets the load balance path by ID. This method can be accessed through the Library.asmx file located in:
	InternalLibrarySearch (see page 1277)	Searches library items. This method can be accessed through the Library.asmx file located in:
	UpdateLibrary (see page 1277)	Updates a library item. This method can be accessed through the Library.asmx file located in:
	UpdateLibrarySettings (see page 1278)	Update the library settings. This method can be accessed through the Library.asmx file located in:
	UpdateLoadBalanceSettings (see page 1278)	Updates the load balance settings. This method can be accessed through the Library.asmx file located in:
	UpdateQlinkTemplates (see page 1278)	Updates the Quicklink template. This method can be accessed through the Library.asmx file located in:

Library.AddLoadBalanceItem Method

Adds a load balance path for files and images.

This method can be accessed through the Library.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Library.asmx`

C#

```
public void AddLoadBalanceItem(string Type, Collection pagedata);
```

Visual Basic

```
Public Function AddLoadBalanceItem(Type As string, pagedata As Collection) As void
```

Parameters

Parameters	Description
string Type	The path type, either images or files.
Collection pagedata	Load balance item values for path and relative.

Remarks

A user with administrator privileges must be logged in to perform this operation.

Library.DeleteLibraryItem Method

Deletes the library item by extension.

This method can be accessed through the Library.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Library.asmx`

C#

```
public void DeleteLibraryItem(Collection pagedata);
```

Visual Basic

```
Public Function DeleteLibraryItem(pagedata As Collection) As void
```

Parameters

Parameters	Description
Collection pagedata	Library (see page 1269) item data.

Remarks

A user with administrator privileges must be logged in to perform this operation.

Library.DeleteLibraryItemById Method

Deletes the library items from the system.

This method can be accessed through the Library.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Library.asmx`

C#

```
public void DeleteLibraryItemById(Collection pagedata);
```

Visual Basic

```
Public Function DeleteLibraryItemById(pagedata As Collection) As void
```

Parameters

Parameters	Description
Collection pagedata	Library (see page 1269) item data.

Remarks

A user with administrator privileges must be logged in to perform this operation.

Library.DeleteLoadBalanceItem Method

Deletes the load balance item from the system.

This method can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteLoadBalanceItem", Description = "")]
public void DeleteLoadBalanceItem(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteLoadBalanceItem", Description = "")>
Public Function DeleteLoadBalanceItem(Id As long) As void
```

Parameters

Parameters	Description
long Id	The load balance item ID.

Remarks

A user with administrator privileges must be logged in to perform this operation.

Library.GetAllChildLibItems Method

Loads all of the library items by type.

This method can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllChildLibItems", Description = "Loads all of the library items
by type")]
public LibraryData[] GetAllChildLibItems(string Type, long ParentId, string OrderBy, int
currentPageNum, int pageSize, ref int totalPages);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllChildLibItems", Description = "Loads all of the library items
by type")>
Public Function GetAllChildLibItems(Type As string, ParentId As long, OrderBy As string,
currentPageNum As Integer, pageSize As Integer, ByRef totalPages As Integer) As LibraryData()
```

Parameters

Parameters	Description
string Type	The type (Required).
long ParentId	The folder's ID.
string OrderBy	The order to load the library items.
int currentPageNum	The current page number.
int pageSize	The page size.
ref int totalPages	The total pages.

Returns

Ektron.Cms.LibraryData()

Remarks

User (see page 1369) login required to perform this operation.

GetAllLoadBalancePaths Method

This is the overview for the GetAllLoadBalancePaths method overload.

Description

This method is overloaded.

- GetAllLoadBalancePaths (Long, String) - returns an array of LoadBalanceData for either images or files.
- GetAllLoadBalancePaths (String) - returns all the load balance paths as a array of LoadBalanceData.

Overload List

	Name	Description
	Library.GetAllLoadBalancePaths (long, string) (see page 1273)	Returns an array of LoadBalanceData. This method can be accessed through the Library.asmx file located in:
	Library.GetAllLoadBalancePaths (string) (see page 1274)	Returns all the load balance paths as a array of LoadBalanceData. This method can be accessed through the Library.asmx file located in:

Library.GetAllLoadBalancePaths Method (long, string)

Returns an array of LoadBalanceData.

This method can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllLoadBalancePaths By Id and Type", Description = "")]
public LoadBalanceData[] GetAllLoadBalancePaths(long Id, string Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllLoadBalancePaths By Id and Type", Description = "")>
Public Function GetAllLoadBalancePaths(Id As long, Type As string) As LoadBalanceData()
```

Parameters

Parameters	Description
long Id	Load balance ID.
string Type	Use images or files.

Returns

Ektron.Cms.LoadBalanceData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

Library.GetAllLoadBalancePaths Method (string)

Returns all the load balance paths as a array of LoadBalanceData.

This method can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllLoadBalancePaths By Type", Description = "")]
public LoadBalanceData[] GetAllLoadBalancePaths(string Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllLoadBalancePaths By Type", Description = "")>
Public Function GetAllLoadBalancePaths(Type As string) As LoadBalanceData()
```

Parameters

Parameters	Description
string Type	Type (images or files).

Returns

Ektron.Cms.LoadBalanceData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

GetLibraryItem Method

This is the overview for the GetLibraryItem method overload.

Description

This method is overloaded.

- GetLibraryItem (Long) - loads the library item data based on the given library ID.
- GetLibraryItem (Long, Long) - loads the library item data based on the given library ID and parent folder ID.

Overload List

	Name	Description
	Library.GetLibraryItem (long) (see page 1274)	Loads the library item data. This method can be accessed through the Library.asmx file located in:
	Library.GetLibraryItem (long, long) (see page 1275)	Loads the library item's data. This method can be accessed through the Library.asmx file located in:

Library.GetLibraryItem Method (long)

Loads the library item data.

This method can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLibraryItem By Id", Description = "")]
public LibraryData GetLibraryItem(long LibID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLibraryItem By Id", Description = "")>
Public Function GetLibraryItem(LibID As long) As LibraryData
```

Parameters

Parameters	Description
long LibID	The library ID.

Returns

Ektron.Cms.LibraryData

Library.GetLibraryItem Method (long, long)

Loads the library item's data.

This method can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLibraryItem By Id And ParentID", Description = "")]
public LibraryData GetLibraryItem(long Id, long ParentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLibraryItem By Id And ParentID", Description = "")>
Public Function GetLibraryItem(Id As long, ParentId As long) As LibraryData
```

Parameters

Parameters	Description
long Id	The library's ID.
long ParentId	The folder's ID.

Returns

Ektron.Cms.LibraryData

Remarks

Read permission to the folder is required to perform this operation.

Library.GetLibrarySettings Method

Loads the library settings.

This method can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLibrarySettings", Description = "")]
public LibraryConfigData GetLibrarySettings(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLibrarySettings", Description = "")>
Public Function GetLibrarySettings(Id As long) As LibraryConfigData
```

Parameters

Parameters	Description
long Id	The folder's ID.

Returns

Ektron.Cms.LibraryConfigData

Remarks

Login (see page 347) required to perform this operation.

Library.GetLibraryTypes Method

Loads the library types.

This method can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLibraryTypes", Description = "")]
public LibraryTypeData[] GetLibraryTypes(long Id, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLibraryTypes", Description = "")>
Public Function GetLibraryTypes(Id As long, OrderBy As string) As LibraryTypeData()
```

Parameters

Parameters	Description
long Id	The folder's ID.
string OrderBy	Order by clause.

Returns

Object

Remarks

Library (see page 1269) read permission is required to perform this operation.

Library.GetLoadBalancePath Method

Gets the load balance path by ID.

This method can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "", Description = "")]
public LoadBalanceData GetLoadBalancePath(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "", Description = "")>
Public Function GetLoadBalancePath(Id As long) As LoadBalanceData
```

Parameters

Parameters	Description
long Id	The load balance path's ID.

Returns

Ektron.Cms.LoadBalanceData

Library.InternalLibrarySearch Method

Searches library items.

This method can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

C#

```
public SearchResultData[] InternalLibrarySearch(Collection pagedata,
Ektron.Cms.Common.SearchAssetRequest sar);
```

Visual Basic

```
Public Function InternalLibrarySearch(pagedata As Collection, sar As
Ektron.Cms.Common.SearchAssetRequest) As SearchResultData()
```

Parameters

Parameters	Description
Collection pagedata	Search (see page 899) criteria.
Ektron.Cms.Common.SearchAssetRequest sar	Search (see page 899) asset criteria.

Returns

Ektron.Cms.SearchResultData()

Remarks

User (see page 1369) login is required to perform this operation.

Library.UpdateLibrary Method

Updates a library item.

This method can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

C#

```
public void UpdateLibrary(Collection pagedata);
```

Visual Basic

```
Public Function UpdateLibrary(pagedata As Collection) As void
```

Parameters

Parameters	Description
Collection pagedata	Library (see page 1269) item data.

Library.UpdateLibrarySettings Method

Update the library settings.

This method can be accessed through the Library.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Library.asmx`

C#

```
public void UpdateLibrarySettings(Collection pagedata);
```

Visual Basic

```
Public Function UpdateLibrarySettings(pagedata As Collection) As void
```

Parameters

Parameters	Description
Collection pagedata	Library (see page 1269) settings data.

Remarks

A user with administrator privileges must be logged in to perform this operation.

Library.UpdateLoadBalanceSettings Method

Updates the load balance settings.

This method can be accessed through the Library.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Library.asmx`

C#

```
public void UpdateLoadBalanceSettings(string Id, Collection pagedata);
```

Visual Basic

```
Public Function UpdateLoadBalanceSettings(Id As string, pagedata As Collection) As void
```

Parameters

Parameters	Description
string Id	The load balance data.
Collection pagedata	The load balance items.

Description

[WebMethod(MessageName = "UpdateLoadBalanceSettings", Description = "")]

Remarks

User (see page 1369) login with administrator privileges are required to perform this operation.

Library.UpdateQlinkTemplates Method

Updates the Quicklink template.

This method can be accessed through the Library.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Library.asmx
```

C#

```
public void UpdateQlinkTemplates(Collection pagedata);
```

Visual Basic

```
Public Function UpdateQlinkTemplates(pagedata As Collection) As void
```

Parameters

Parameters	Description
Collection pagedata	Quicklink template data.

Remarks

A user with administrator privileges must be logged in to perform this operation.

MessageBoard Class

This section describes the Web Service methods used with Ektron's Message Board feature. This feature is a part Ektron's Community (see page 750) Platform. The MessageBoard Web service is a wrapper for the MessageBoard API (see page 669): `Ektron.Cms.API.Community.MessageBoard` (see page 787). These methods can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy

C#

```
public class MessageBoard : EktronBaseService;
```

Visual Basic

```
Public Class MessageBoard
Inherits EktronBaseService
```

File

MessageBoard.cs

MessageBoard Methods

The methods of the MessageBoard class are listed here.

Public Methods

	Name	Description
	AddMessageBoardEntry (see page 1280)	Adds a Message Board Entry. This method can be accessed through the MessageBoard.asmx file located in:
	ApproveMessage (see page 1280)	This is the overview for the ApproveMessage method overload.
	DeleteMessageBoardEntries (see page 1283)	This is the overview for the DeleteMessageBoardEntries method overload.
	DeleteMessageBoardEntriesByUser (see page 1285)	This is the overview for the DeleteMessageBoardEntriesByUser method overload.
	DeleteMessageBoardEntry (see page 1287)	This is the overview for the DeleteMessageBoardEntry method overload.

	GetAllMessageBoardEntries (see page 1289)	This is the overview for the GetAllMessageBoardEntries method overload.
	GetAllMessageBoardEntriesByUser (see page 1292)	This is the overview for the GetAllMessageBoardEntriesByUser method overload.
	GetMessageBoardEntry (see page 1294)	This is the overview for the GetMessageBoardEntry method overload.
	IsUserMessageBoardAdmin (see page 1297)	This function determines if the user has administrative privileges for the message board. This method can be accessed through the MessageBoard.asmx file located in:
	UpdateMessageText (see page 1297)	This is the overview for the UpdateMessageText method overload.

MessageBoard.AddMessageBoardEntry Method

Adds a Message Board Entry.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddMessageBoardEntry", Description = "Adds a Message Board Entry.")]
public MessageBoardData AddMessageBoardEntry(long objectId,
EkEnumeration.MessageBoardObjectType objectType, string messageText, bool isModerated, long
callerUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddMessageBoardEntry", Description = "Adds a Message Board Entry.")>
Public Function AddMessageBoardEntry(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, messageText As string, isModerated As bool, callerUserId
As long) As MessageBoardData
```

Parameters

Parameters	Description
long objectId	The ID of the object associated with the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.
string messageText	Text to be added an entry.
bool isModerated	Set to True to if the message board is moderated.
long callerUserId	The user ID of the member requesting data.

ApproveMessage Method

This is the overview for the ApproveMessage method overload.

Description

This method is overloaded.

- ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType) - change the flag of the message from pending to approved by providing the message ID, object ID and type.
- ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType, Long) - change the flag of the message from pending to approved by providing the message ID, object ID and type, and the ID of the user who changing the flag.
- ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Bool) - change the flag of the message from pending to approved by providing the message ID, object ID and type, the ID of the user who changing the flag and whether the message board is moderated.

Overload List

	Name	Description
	MessageBoard.ApproveMessage (long, long, EkEnumeration.MessageBoardObjectType) (see page 1281)	This function changes the flag of the message from pending to approve. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.ApproveMessage (long, long, EkEnumeration.MessageBoardObjectType, long) (see page 1281)	This function changes the flag of the message from pending to approve. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.ApproveMessage (long, long, EkEnumeration.MessageBoardObjectType, long, bool) (see page 1282)	This function changes the flag of the message from pending to approve. This method can be accessed through the MessageBoard.asmx file located in:

MessageBoard.ApproveMessage Method (long, long, EkEnumeration.MessageBoardObjectType)

This function changes the flag of the message from pending to approve.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ApproveMessage", Description = "Updates the Text of a message.")]
public MessageBoardData ApproveMessage(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ApproveMessage", Description = "Updates the Text of a message.")>
Public Function ApproveMessage(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType) As MessageBoardData
```

Parameters

Parameters	Description
long messageId	The unique ID of the message.
long objectId	The ID of the object.
EkEnumeration.MessageBoardObjectType objectType	The type of the object.

Returns

Returns the updated row.

MessageBoard.ApproveMessage Method (long, long, EkEnumeration.MessageBoardObjectType, long)

This function changes the flag of the message from pending to approve.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
```

```
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ApproveMessage with caller user id", Description = "Updates the Text
of a message.")]
public MessageBoardData ApproveMessage(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, long callerUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ApproveMessage with caller user id", Description = "Updates the Text
of a message.")>
Public Function ApproveMessage(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long) As MessageBoardData
```

Parameters

Parameters	Description
long messageId	The unique ID of the message.
long objectId	The ID of the object.
EkEnumeration.MessageBoardObjectType objectType	The type of the object.
long callerUserId	The user ID of the member doing the update.

Returns

Returns the updated row.

MessageBoard.ApproveMessage Method (long, long, EkEnumeration.MessageBoardObjectType, long, bool)

This function changes the flag of the message from pending to approve.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ApproveMessage with caller user id and moderated flag", Description
= "Updates the Text of a message.")]
public MessageBoardData ApproveMessage(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, long callerUserId, bool isModerated);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ApproveMessage with caller user id and moderated flag", Description
= "Updates the Text of a message.")>
Public Function ApproveMessage(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long, isModerated As bool) As
MessageBoardData
```

Parameters

Parameters	Description
long messageId	The unique ID of the message.
long objectId	The ID of the object.
EkEnumeration.MessageBoardObjectType objectType	The type of the object.
long callerUserId	The user ID of the member doing the update.
bool isModerated	The moderation switch.

Returns

Returns the updated row.

DeleteMessageBoardEntries Method

This is the overview for the DeleteMessageBoardEntries method overload.

Description

This method is overloaded.

- DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType) - deletes all the message board records for a particular object based on the object ID and type.
- DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, Long)Ektron.Services.MessageBoard.DeleteMessageBoardEntries@int@EkEnumeration.MessageBoardObjectType@int - deletes all the message board records for a particular object based on the object ID and type, and the user ID of the member deleting the messages.
- DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, Long, Bool) - deletes all the message board records for a particular object based on the object ID and type, the user ID of the member deleting the messages, and whether the message board is moderated.

Overload List

	Name	Description
	MessageBoard.DeleteMessageBoardEntries (long, EkEnumeration.MessageBoardObjectType) (see page 1283)	Deletes all the message board records for a particular object. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.DeleteMessageBoardEntries (long, EkEnumeration.MessageBoardObjectType, long) (see page 1284)	Deletes all the message board records for a particular object. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.DeleteMessageBoardEntries (long, EkEnumeration.MessageBoardObjectType, long, bool) (see page 1284)	Deletes all the message board records for a particular object. This method can be accessed through the MessageBoard.asmx file located in:

MessageBoard.DeleteMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType)

Deletes all the message board records for a particular object.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntries", Description = "Deletes all the message
board records for a particular object.")]
public bool DeleteMessageBoardEntries(long objectId, EkEnumeration.MessageBoardObjectType
objectType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntries", Description = "Deletes all the message
board records for a particular object.")>
Public Function DeleteMessageBoardEntries(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType) As bool
```

Parameters

Parameters	Description
long objectId	The ID of the object that has the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.

Returns

Returns true when all the messages are deleted successfully.

MessageBoard.DeleteMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, long)

Deletes all the message board records for a particular object.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntries with caller user id", Description =
"Deletes all the message board records for a particular object.")]
public bool DeleteMessageBoardEntries(long objectId, EkEnumeration.MessageBoardObjectType
objectType, long callerUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntries with caller user id", Description =
"Deletes all the message board records for a particular object.")>
Public Function DeleteMessageBoardEntries(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long) As bool
```

Parameters

Parameters	Description
long objectId	The ID of the object associated with the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.
long callerUserId	The user ID of the member deleting the messages.

Returns

Returns true when all the messages are deleted successfully

MessageBoard.DeleteMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, long, bool)

Deletes all the message board records for a particular object.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntries with caller user id and moderated flag",
Description = "Deletes all the message board records for a particular object.")]
public bool DeleteMessageBoardEntries(long objectId, EkEnumeration.MessageBoardObjectType
objectType, long callerUserId, bool isModerated);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntries with caller user id and moderated flag",
Description = "Deletes all the message board records for a particular object.")>
Public Function DeleteMessageBoardEntries(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long, isModerated As bool) As bool
```

Parameters

Parameters	Description
long objectId	The ID of the object that has the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.
long callerUserId	The user ID of the member deleting the messages.
bool isModerated	The moderation switch.

Returns

Returns true when all the messages are deleted successfully.

DeleteMessageBoardEntriesByUser Method

This is the overview for the DeleteMessageBoardEntriesByUser method overload.

Description

This method is overloaded.

- DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long) - deletes all of a user's messages on a given message board based on the object ID and type, and the user's ID.
- DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, Long, Bool) - deletes all of a user's messages on a given message board based on the object ID and type, the ID of the user whose messages are being removed and the ID of the user removing the messages.
- DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, Long, Bool) - deletes all of a user's messages on a given message board based on the object ID and type, the ID of the user whose messages are being removed, the ID of the user removing the messages and whether the message board is moderated.

Overload List

	Name	Description
	MessageBoard.DeleteMessageBoardEntriesByUser (long, EkEnumeration.MessageBoardObjectType, long) (see page 1285)	Deletes all the message board records for a specified user. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.DeleteMessageBoardEntriesByUser (long, EkEnumeration.MessageBoardObjectType, long, long) (see page 1286)	Deletes all the message board records for a specified user. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.DeleteMessageBoardEntriesByUser (long, EkEnumeration.MessageBoardObjectType, long, long, bool) (see page 1287)	Deletes all the message board records for a specified user. This method can be accessed through the MessageBoard.asmx file located in:

MessageBoard.DeleteMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long)

Deletes all the message board records for a specified user.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntriesByUser", Description = "Deletes all the
message board records for a particular User.")]
public bool DeleteMessageBoardEntriesByUser(long objectId,
EkEnumeration.MessageBoardObjectType objectType, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntriesByUser", Description = "Deletes all the
message board records for a particular User.")>
Public Function DeleteMessageBoardEntriesByUser(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long) As bool
```

Parameters

Parameters	Description
long objectId	The ID of the object associated with the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.
long userId	The user ID whose messages to you want to delete.

MessageBoard.DeleteMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, long)

Deletes all the message board records for a specified user.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntriesByUser with caller user ID", Description =
"Deletes all the message board records for a particular User.")]
public bool DeleteMessageBoardEntriesByUser(long objectId,
EkEnumeration.MessageBoardObjectType objectType, long userId, long callerUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntriesByUser with caller user ID", Description =
"Deletes all the message board records for a particular User.")>
Public Function DeleteMessageBoardEntriesByUser(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long, callerUserId As long) As bool
```

Parameters

Parameters	Description
long objectId	The ID of the object that has the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.
long userId	The user ID whose messages to you want to delete.
long callerUserId	The user ID of the member deleting the messages.

MessageBoard.DeleteMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, long, bool)

Deletes all the message board records for a specified user.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntriesByUser with caller user ID and moderated
flag", Description = "Deletes all the message board records for a particular User.")]
public bool DeleteMessageBoardEntriesByUser(long objectId,
EkEnumeration.MessageBoardObjectType objectType, long userId, long callerUserId, bool
isModerated);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntriesByUser with caller user ID and moderated
flag", Description = "Deletes all the message board records for a particular User.")>
Public Function DeleteMessageBoardEntriesByUser(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long, callerUserId As long, isModerated As
bool) As bool
```

Parameters

Parameters	Description
long objectId	The ID of the object associated with the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.
long userId	The user ID whose messages to you want to delete.
long callerUserId	The user ID of the member deleting the messages.
bool isModerated	The moderation switch.

DeleteMessageBoardEntry Method

This is the overview for the DeleteMessageBoardEntry method overload.

Description

This method is overloaded.

- DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType) - deletes single message board record based on the message ID, object ID and object type.
- DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long) - deletes single message board record based on the message ID, object ID, object type and user ID requesting the data.
- DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Bool) - deletes single message board record based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

Overload List

	Name	Description
	MessageBoard.DeleteMessageBoardEntry (long, long, EkEnumeration.MessageBoardObjectType) (see page 1288)	Deletes single message board record. This method can be accessed through the MessageBoard.asmx file located in:

	MessageBoard.DeleteMessageBoardEntry (long, long, EkEnumeration.MessageBoardObjectType, long) (see page 1288)	Deletes single message board record. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.DeleteMessageBoardEntry (long, long, EkEnumeration.MessageBoardObjectType, long, bool) (see page 1289)	Deletes single message board record. This method can be accessed through the MessageBoard.asmx file located in:

MessageBoard.DeleteMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType)

Deletes single message board record.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntry", Description = "Deletes single message
board record.")]
public bool DeleteMessageBoardEntry(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntry", Description = "Deletes single message
board record.")>
Public Function DeleteMessageBoardEntry(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType) As bool
```

Parameters

Parameters	Description
long messageId	The unique ID of the message.
long objectId	The ID of the object associated with the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.

Returns

Returns true when the message is deleted successfully.

MessageBoard.DeleteMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long)

Deletes single message board record.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntry with caller user id", Description = "Deletes
single message board record.")]
public bool DeleteMessageBoardEntry(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, long callerUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntry with caller user id", Description = "Deletes
single message board record.")>
Public Function DeleteMessageBoardEntry(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long) As bool
```

Parameters

Parameters	Description
long messageId	The unique ID of the message.
long objectId	The ID of the object that has the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.
long callerUserId	The user ID of the member deleting the message.

Returns

Returns true when the message is deleted successfully.

MessageBoard.DeleteMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long, bool)

Deletes single message board record.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntry with caller user id and moderated flag",
Description = "Deletes single message board record.")]
public bool DeleteMessageBoardEntry(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, long callerUserId, bool isModerated);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntry with caller user id and moderated flag",
Description = "Deletes single message board record.")>
Public Function DeleteMessageBoardEntry(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long, isModerated As bool) As bool
```

Parameters

Parameters	Description
long messageId	The unique ID of the message.
long objectId	The ID of the object associated with the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.
long callerUserId	The user ID of the member deleting the message.
bool isModerated	The moderation switch.

Returns

Returns true when the message is deleted successfully.

GetAllMessageBoardEntries Method

This is the overview for the GetAllMessageBoardEntries method overload.

Description

This method is overloaded.

- GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo) - gets all messages for a particular object based on the object ID, object type and paging information.
- GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long) - gets all messages for a particular object based on the object ID, object type, ID of the user requesting the list and paging information.
- GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long, Bool) - gets all messages for a particular object based on the object ID, object type, ID of the user requesting the list, paging information and the moderation switch.

Overload List

	Name	Description
	MessageBoard.GetAllMessageBoardEntries (long, EkEnumeration.MessageBoardObjectType, PagingInfo) (see page 1290)	Gets all messages for a particular object. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.GetAllMessageBoardEntries (long, EkEnumeration.MessageBoardObjectType, PagingInfo, long) (see page 1291)	Gets all messages for a particular object. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.GetAllMessageBoardEntries (long, EkEnumeration.MessageBoardObjectType, PagingInfo, long, bool) (see page 1291)	Gets all messages for a particular object. This method can be accessed through the MessageBoard.asmx file located in:

MessageBoard.GetAllMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, PagingInfo)

Gets all messages for a particular object.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllMessageBoardEntries", Description = "Gets all messages for a
particular object. ")]
public Collection<MessageBoardData> GetAllMessageBoardEntries(long objectId,
EkEnumeration.MessageBoardObjectType objectType, PagingInfo paging);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllMessageBoardEntries", Description = "Gets all messages for a
particular object. ")>
Public Function GetAllMessageBoardEntries(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, paging As PagingInfo) As Collection<MessageBoardData>
```

Parameters

Parameters	Description
long objectId	The ID of the object.

EkEnumeration.MessageBoardObjectType objectType	The type of the object.
PagingInfo paging	The paging object. Setting "RecordsPerPage" to zero retrieves all rows.

Returns

Returns a collection of message board data objects for each record retrieved from the database.

MessageBoard.GetAllMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, PagingInfo, long)

Gets all messages for a particular object.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllMessageBoardEntries with caller user id", Description = "Gets
all messages for a particular object. ")]
public Collection<MessageBoardData> GetAllMessageBoardEntries(long objectId,
EkEnumeration.MessageBoardObjectType objectType, PagingInfo paging, long callerUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllMessageBoardEntries with caller user id", Description = "Gets
all messages for a particular object. ")>
Public Function GetAllMessageBoardEntries(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, paging As PagingInfo, callerUserId As long) As
Collection<MessageBoardData>
```

Parameters

Parameters	Description
long objectId	The ID of the object.
EkEnumeration.MessageBoardObjectType objectType	The type of the object.
PagingInfo paging	The paging object. Setting "RecordsPerPage" to zero retrieves all rows.
long callerUserId	The user ID of the member requesting data.

Returns

Returns a collection of message board data objects for each record retrieved from the database.

MessageBoard.GetAllMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, PagingInfo, long, bool)

Gets all messages for a particular object.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllMessageBoardEntries with caller user id and moderated flag",
Description = "Gets all messages for a particular object. ")]
public Collection<MessageBoardData> GetAllMessageBoardEntries(long objectId,
EkEnumeration.MessageBoardObjectType objectType, PagingInfo paging, long callerUserId, bool
isModerated);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllMessageBoardEntries with caller user id and moderated flag",
Description = "Gets all messages for a particular object. ")>
Public Function GetAllMessageBoardEntries(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, paging As PagingInfo, callerUserId As long, isModerated
As bool) As Collection<MessageBoardData>
```

Parameters

Parameters	Description
long objectId	The ID of the object.
EkEnumeration.MessageBoardObjectType objectType	The type of the object.
PagingInfo paging	The paging object. Setting "RecordsPerPage" to zero retrieves all rows.
long callerUserId	The user ID of the member requesting data.
bool isModerated	The moderation switch.

Returns

Returns a collection of message board data objects for each record retrieved from the database.

GetAllMessageBoardEntriesByUser Method

This is the overview for the GetAllMessageBoardEntriesByUser method overload.

Description

This method is overloaded.

- GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo) - gets all messages entered by a user on a given message board based on the object ID and type, the user's ID and paging information.
- GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long) - gets all messages entered by a user on a given message board based on the object ID and type, the ID of the user whose messages are being removed, the ID of the user removing the messages and paging information.
- GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long, Bool) - gets all messages entered by a user on a given message board based on the object ID and type, the ID of the user whose messages are being removed, the ID of the user removing the messages and whether the message board is moderated.

Overload List

	Name	Description
	MessageBoard.GetAllMessageBoardEntriesByUser (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo) (see page 1293)	Gets all messages entered by a user. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.GetAllMessageBoardEntriesByUser (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo, long) (see page 1293)	Gets all messages entered by a user. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.GetAllMessageBoardEntriesByUser (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo, long, bool) (see page 1294)	Gets all messages entered by a user. This method can be accessed through the MessageBoard.asmx file located in:

MessageBoard.GetAllMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo)

Gets all messages entered by a user.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllMessageBoardEntriesByUser", Description = "Gets all messages
for a particular user. ")]
public Collection<MessageBoardData> GetAllMessageBoardEntriesByUser(long objectId,
EkEnumeration.MessageBoardObjectType objectType, long userId, PagingInfo paging);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllMessageBoardEntriesByUser", Description = "Gets all messages
for a particular user. ")>
Public Function GetAllMessageBoardEntriesByUser(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long, paging As PagingInfo) As
Collection<MessageBoardData>
```

Parameters

Parameters	Description
long objectId	The ID of the object.
EkEnumeration.MessageBoardObjectType objectType	The type of the object.
long userId	The user ID for which to retrieve data.
PagingInfo paging	The paging object. Setting "RecordsPerPage" to zero retrieves all rows.

Returns

Returns a collection of message board data objects for each record retrieved from the database.

MessageBoard.GetAllMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo, long)

Gets all messages entered by a user.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllMessageBoardEntriesByUser with caller user id", Description =
"Gets all messages for a particular user. ")]
public Collection<MessageBoardData> GetAllMessageBoardEntriesByUser(long objectId,
EkEnumeration.MessageBoardObjectType objectType, long userId, PagingInfo paging, long
callerUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllMessageBoardEntriesByUser with caller user id", Description =
"Gets all messages for a particular user. ")>
Public Function GetAllMessageBoardEntriesByUser(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long, paging As PagingInfo, callerUserId As
```

```
long) As Collection<MessageBoardData>
```

Parameters

Parameters	Description
long objectId	The ID of the object.
EkEnumeration.MessageBoardObjectType objectType	The type of the object.
long userId	The user ID for which to retrieve data.
PagingInfo paging	The paging object. Setting "RecordsPerPage" to zero retrieves all rows.
long callerUserId	The user ID of the member requesting data.

Returns

Returns a collection of message board data objects for each record retrieved from the database.

MessageBoard.GetAllMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo, long, bool)

Gets all messages entered by a user.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllMessageBoardEntriesByUser with caller user id and moderated
flag", Description = "Gets all messages for a particular user. ")]
public Collection<MessageBoardData> GetAllMessageBoardEntriesByUser(long objectId,
EkEnumeration.MessageBoardObjectType objectType, long userId, PagingInfo paging, long
callerUserId, bool isModerated);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllMessageBoardEntriesByUser with caller user id and moderated
flag", Description = "Gets all messages for a particular user. ")>
Public Function GetAllMessageBoardEntriesByUser(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long, paging As PagingInfo, callerUserId As
long, isModerated As bool) As Collection<MessageBoardData>
```

Parameters

Parameters	Description
long objectId	The ID of the object.
EkEnumeration.MessageBoardObjectType objectType	The type of the object.
long userId	The user ID for which to retrieve data for.
PagingInfo paging	The paging object. Setting "RecordsPerPage" to zero retrieves all rows.
long callerUserId	The user ID of the member requesting data.
bool isModerated	The moderation switch.

Returns

Returns a collection of message board data objects for each record retrieved from the database.

GetMessageBoardEntry Method

This is the overview for the GetMessageBoardEntry method overload.

Description

This method is overloaded.

- GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType) - gets a specific message from a message board based on the message ID, object ID and object type.
- GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long) - gets a specific message from a message board based on the message ID, object ID, object type and user ID requesting the data.
- GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Bool) - gets a specific message from a message board based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

Overload List

	Name	Description
	MessageBoard.GetMessageBoardEntry (long, long, EkEnumeration.MessageBoardObjectType) (see page 1295)	Returns a single row for the requested parameters. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.GetMessageBoardEntry (long, long, EkEnumeration.MessageBoardObjectType, long) (see page 1296)	Returns a single row for the requested parameters. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.GetMessageBoardEntry (long, long, EkEnumeration.MessageBoardObjectType, long, bool) (see page 1296)	Returns a single row for the requested parameters. This method can be accessed through the MessageBoard.asmx file located in:

MessageBoard.GetMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType)

Returns a single row for the requested parameters.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMessageBoardEntry", Description = "Returns a single row for the
requested parameters.")]
public MessageBoardData GetMessageBoardEntry(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMessageBoardEntry", Description = "Returns a single row for the
requested parameters.")>
Public Function GetMessageBoardEntry(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType) As MessageBoardData
```

Parameters

Parameters	Description
long messageId	The unique ID of the message.
long objectId	The ID of the object associated with the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.

Returns

The message board data object with the data from the database.

MessageBoard.GetMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long)

Returns a single row for the requested parameters.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMessageBoardEntry with caller user id", Description = "Returns a
single row for the requested parameters.")]
public MessageBoardData GetMessageBoardEntry(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, long callerUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMessageBoardEntry with caller user id", Description = "Returns a
single row for the requested parameters.")>
Public Function GetMessageBoardEntry(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long) As MessageBoardData
```

Parameters

Parameters	Description
long messageId	The unique ID of the message.
long objectId	The ID of the object associated with the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.
long callerUserId	The user ID of the member requesting data.

Returns

The message board data object with the data from the database.

MessageBoard.GetMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long, bool)

Returns a single row for the requested parameters.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMessageBoardEntry with caller user id and moderated flag",
Description = "Returns a single row for the requested parameters.")]
public MessageBoardData GetMessageBoardEntry(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, long callerUserId, bool isModerated);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMessageBoardEntry with caller user id and moderated flag",
Description = "Returns a single row for the requested parameters.")>
Public Function GetMessageBoardEntry(messageId As long, objectId As long, objectType As
```

```
EkEnumeration.MessageBoardObjectType, callerUserId As long, isModerated As bool) As
MessageBoardData
```

Parameters

Parameters	Description
long objectId	The ID of the object associated with the message board.
EkEnumeration.MessageBoardObjectType objectType	The type of object.
long callerUserId	The user ID of the member requesting data.
bool isModerated	The moderation switch.
MessageId	The unique ID of the message.

Returns

The message board data object with the data from the database.

MessageBoard.IsUserMessageBoardAdmin Method

This function determines if the user has administrative privileges for the message board.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsUserMessageBoardAdmin", Description = "This function determines if
the user has administrative privileges for the message board.")]
public bool IsUserMessageBoardAdmin(long objectId, EkEnumeration.MessageBoardObjectType
objectType, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsUserMessageBoardAdmin", Description = "This function determines if
the user has administrative privileges for the message board.")>
Public Function IsUserMessageBoardAdmin(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long) As bool
```

Parameters

Parameters	Description
long userId	The user ID of the user

Returns

Returns true if the user has administrative privileges.

UpdateMessageText Method

This is the overview for the UpdateMessageText method overload.

Description

This method is overloaded.

- UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String) - updates the text of a message based on the message ID, object ID and type, and the new text.
- UpdatedMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long) - updates the text of a message based on the message ID, object ID and type, the new text and the ID of the user changing the text.
- UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long, Bool) - updates the text of a message based on the message ID, object ID and type, the new text, the ID of the user changing the text and whether the

board is moderated.

Overload List

	Name	Description
	MessageBoard.UpdateMessageText (long, long, EkEnumeration.MessageBoardObjectType, string) (see page 1298)	Updates the Text of a message. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.UpdateMessageText (long, long, EkEnumeration.MessageBoardObjectType, string, long) (see page 1298)	Updates the Text of a message. This method can be accessed through the MessageBoard.asmx file located in:
	MessageBoard.UpdateMessageText (long, long, EkEnumeration.MessageBoardObjectType, string, long, bool) (see page 1299)	Updates the Text of a message. This method can be accessed through the MessageBoard.asmx file located in:

MessageBoard.UpdateMessageText Method (long, long, EkEnumeration.MessageBoardObjectType, string)

Updates the Text of a message.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateMessageText", Description = "Updates the Text of a message.")]
public MessageBoardData UpdateMessageText(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, string messageText);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateMessageText", Description = "Updates the Text of a message.")>
Public Function UpdateMessageText(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, messageText As string) As MessageBoardData
```

Parameters

Parameters	Description
long messageId	The unique ID of the message.
long objectId	The ID of the object.
EkEnumeration.MessageBoardObjectType objectType	The type of the object.
string messageText	The new message text.

Returns

Returns the updated row.

MessageBoard.UpdateMessageText Method (long, long, EkEnumeration.MessageBoardObjectType, string, long)

Updates the Text of a message.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateMessageText with caller user id", Description = "Updates the
Text of a message.")]
public MessageBoardData UpdateMessageText(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, string messageText, long callerUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateMessageText with caller user id", Description = "Updates the
Text of a message.")>
Public Function UpdateMessageText(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, messageText As string, callerUserId As long) As
MessageBoardData
```

Parameters

Parameters	Description
long messageId	The unique ID of the message.
long objectId	The ID of the object.
EkEnumeration.MessageBoardObjectType objectType	The type of the object.
string messageText	The new message text.
long callerUserId	The user ID of the member doing the update.

Returns

Returns the updated row.

MessageBoard.UpdateMessageText Method (long, long, EkEnumeration.MessageBoardObjectType, string, long, bool)

Updates the Text of a message.

This method can be accessed through the MessageBoard.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/MessageBoard.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateMessageText with caller user id and moderated flag",
Description = "Updates the Text of a message.")]
public MessageBoardData UpdateMessageText(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, string messageText, long callerUserId, bool
isModerated);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateMessageText with caller user id and moderated flag",
Description = "Updates the Text of a message.")>
Public Function UpdateMessageText(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, messageText As string, callerUserId As long, isModerated
As bool) As MessageBoardData
```

Parameters

Parameters	Description
long messageId	The unique ID of the message.

long objectId	The ID of the object.
EkEnumeration.MessageBoardObjectType objectType	The type of the object.
string messageText	The new message text.
long callerUserId	The user ID of the member doing the update.
bool isModerated	The moderation switch.

Returns

Returns the updated row.

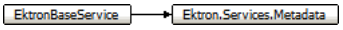
Metadata Class

This section describes the Web Service methods used with Metadata. The Metadata Web service is a wrapper for the Metadata API (see page 669): `Ektron.Cms.API.Metadata` (see page 714). These methods can be accessed through the Metadata.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Metadata.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Metadata : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Metadata
Inherits EktronBaseService
```

File

Metadata.cs

Metadata Methods

The methods of the Metadata class are listed here.

Public Methods

	Name	Description
	AddMetaDataType (see page 1301)	Adds a new metadata type to the system. This method can be accessed through the Metadata.asmx file located in:
	DeleteMetadataType (see page 1301)	Deletes a metadata type from the system. This method can be accessed through the Metadata.asmx file located in:
	GetContentMetadataList (see page 1301)	This is the overview for the GetContentMetadataList method overload.
	GetMetadataType (see page 1303)	Loads all content metadata. This method can be accessed through the Metadata.asmx file located in:
	GetMetaDataTypes (see page 1303)	Loads all the metadata types. This method can be accessed through the Metadata.asmx file located in:
	UpdateMetadataType (see page 1304)	Updates the metadata type.

Metadata.AddMetaDataType Method

Adds a new metadata type to the system.

This method can be accessed through the Metadata.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Metadata.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddMetaDataType", Description = "Adds a new metadata type
definition")]
public void AddMetaDataType(ref ContentMetaData metaDataItem);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddMetaDataType", Description = "Adds a new metadata type
definition")>
Public Function AddMetaDataType(ByRef metaDataItem As ContentMetaData) As void
```

Parameters

Parameters	Description
ref ContentMetaData metaDataItem	ContentMetadata object containing metadata definition details

Remarks

A user with administrator privileges must be logged in to perform this operation.

Metadata.DeleteMetadataType Method

Deletes a metadata type from the system.

This method can be accessed through the Metadata.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Metadata.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMetadataType", Description = "")]
public void DeleteMetadataType(long MetadataId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMetadataType", Description = "")>
Public Function DeleteMetadataType(MetadataId As long) As void
```

Parameters

Parameters	Description
long MetadataId	Metatype ID

Remarks

A user with administrator privileges must be logged in to perform this operation.

GetContentMetadataList Method

This is the overview for the GetContentMetadataList method overload.

Description

This method is overloaded.

- GetContentMetadataList (Long) - loads all of the metadata for the given content based on the content's ID.
- GetContentMetadataList (Long, Bool) - loads all of the metadata for the given content base on the content ID. This method allows you to load metadata for a staged content item.

Overload List

	Name	Description
	Metadata.GetContentMetadataList (long) (see page 1302)	Loads all of the metadata for the given content. This method can be accessed through the Metadata.asmx file located in:
	Metadata.GetContentMetadataList (long, bool) (see page 1302)	Loads all of the metadata for the given content. This method can be accessed through the Metadata.asmx file located in:

Metadata.GetContentMetadataList Method (long)

Loads all of the metadata for the given content.

This method can be accessed through the Metadata.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Metadata.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentMetadataList", Description = "")]
public CustomAttributeList GetContentMetadataList(long ContentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentMetadataList", Description = "")>
Public Function GetContentMetadataList(ContentId As long) As CustomAttributeList
```

Parameters

Parameters	Description
long ContentId	The contents ID.

Returns

Ektron.Cms.CustomAttributeList

Remarks

Read permissions required.

Metadata.GetContentMetadataList Method (long, bool)

Loads all of the metadata for the given content.

This method can be accessed through the Metadata.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Metadata.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "GetContentMetadataList with preview", Description = "")]
public CustomAttributeList GetContentMetadataList(long ContentId, bool preview);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentMetadataList with preview", Description = "")>
Public Function GetContentMetadataList(ContentId As long, preview As bool) As
CustomAttributeList
```

Parameters

Parameters	Description
long ContentId	The contents ID.
bool preview	True implies get metadata from staging content, else use published content

Returns

Ektron.Cms.CustomAttributeList

Remarks

Read permissions required.

Metadata.GetMetadataType Method

Loads all content metadata.

This method can be accessed through the Metadata.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Metadata.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMetadataType", Description = "")]
public ContentMetaData GetMetadataType(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMetadataType", Description = "")>
Public Function GetMetadataType(Id As long) As ContentMetaData
```

Parameters

Parameters	Description
long Id	The metadata type ID

Returns

Ektron.Cms.ContentMetaData

Remarks

A user with administrator privileges must be logged in to perform this operation.

Metadata.GetMetaDataTypes Method

Loads all the metadata types.

This method can be accessed through the Metadata.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Metadata.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMetaDataTypes", Description = "")]
public ContentMetaData[] GetMetaDataTypes(string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMetaDataTypes", Description = "")>
Public Function GetMetaDataTypes(OrderBy As string) As ContentMetaData()
```

Parameters

Parameters	Description
string OrderBy	Order by clause

Returns

Ektron.Cms.ContentMetaData()

Remarks

User (see page 1369) login is required to perform this operation.

Metadata.UpdateMetadataType Method

Updates the metadata type.

C#

```
public void UpdateMetadataType(Collection pagedata);
```

Visual Basic

```
Public Function UpdateMetadataType(pagedata As Collection) As void
```

Parameters

Parameters	Description
Collection pagedata	Metadata (see page 1300) types

Remarks

A user with administrator privileges must be logged in to perform this operation.

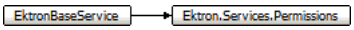
Permissions Class

This section describes the Web Service methods used with Permissions. The Permissions Web service is a wrapper for the Permissions API (see page 669): `Ektron.Cms.API.Permissions` (see page 721). These methods can be accessed through the Permissions.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Permissions.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
```

```
public class Permissions : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Permissions
Inherits EktronBaseService
```

File

Permissions.cs

Permissions Methods

The methods of the Permissions class are listed here.

Public Methods

	Name	Description
	AddItemPermission (see page 1305)	Adds the item permissions for a user or a usergroup. This method can be accessed through the Permissions.asmx file located in:
	DeleteItemPermission (see page 1306)	Deletes a user or a group from the item's permissions. This method can be accessed through the Permissions.asmx file located in:
	DisableContentInheritance (see page 1306)	Disable permissions inheritance on the content and copies current permissions to the content. This method can be accessed through the Permissions.asmx file located in:
	DisableFolderInheritance (see page 1307)	Disable permissions inheritance on the folder and copies current permissions to the folder. This method can be accessed through the Permissions.asmx file located in:
	EnableContentInheritance (see page 1307)	Deletes current permissions from the content and enable permissions inheritance on the content. This method can be accessed through the Permissions.asmx file located in:
	EnableFolderInheritance (see page 1308)	Deletes current permissions from the folder and enable permissions inheritance on the folder. This method can be accessed through the Permissions.asmx file located in:
	GetUserPermissions (see page 1308)	This is the overview for the GetUserPermissions method overload.
	LoadPermissions (see page 1310)	Loads the permission for given object, such as content, folder, task, etc. This method can be accessed through the Permissions.asmx file located in:
	UpdateItemPermission (see page 1310)	Updates the item permissions for a user or a usergroup. This method can be accessed through the Permissions.asmx file located in:

Permissions.AddItemPermission Method

Adds the item permissions for a user or a usergroup.

This method can be accessed through the Permissions.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Permissions.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddItemPermission", Description = "Adds the item permissions for a
user or a usergroup.")]
public bool AddItemPermission(UserPermissionData userPermData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddItemPermission", Description = "Adds the item permissions for a
user or a usergroup.")>
Public Function AddItemPermission(userPermData As UserPermissionData) As bool
```

Parameters

Parameters	Description
UserPermissionData userPermData	User (see page 1369)'s Permissions (see page 1304) Data.

Returns

False if it's successful.

Permissions.DeleteItemPermission Method

Deletes a user or a group from the item's permissions.

This method can be accessed through the Permissions.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Permissions.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteItemPermission", Description = "Deletes a user or a group from
the item's permissions")]
public bool DeleteItemPermission(long id, bool IsGroupId, long itemId,
EkEnumeration.CMSObjectTypes itemType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteItemPermission", Description = "Deletes a user or a group from
the item's permissions")>
Public Function DeleteItemPermission(id As long, IsGroupId As bool, itemId As long, itemType
As EkEnumeration.CMSObjectTypes) As bool
```

Parameters

Parameters	Description
long id	The user or group ID.
bool IsGroupId	Enter true if the ID is a group ID.
long itemId	Enter a content ID or folder ID.
EkEnumeration.CMSObjectTypes itemType	Is the given item ID EkEnumeration.CMSObjectTypes.Content (see page 1198) or EkEnumeration.CMSObjectTypes.Folder (see page 1236)?

Returns

False if it's successful.

Permissions.DisableContentInheritance Method

Disable permissions inheritance on the content and copies current permissions to the content.

This method can be accessed through the Permissions.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Permissions.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DisableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool DisableContentInheritance(long id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "DisableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")>
Public Function DisableContentInheritance(id As long) As bool
```

Parameters

Parameters	Description
long id	The content's ID.

Returns

False if it's successful.

Permissions.DisableFolderInheritance Method

Disable permissions inheritance on the folder and copies current permissions to the folder.

This method can be accessed through the Permissions.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Permissions.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DisableFolderInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool DisableFolderInheritance(long id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DisableFolderInheritance", Description = "Deletes a user or a group
from the item's permissions")>
Public Function DisableFolderInheritance(id As long) As bool
```

Parameters

Parameters	Description
long id	The Folder (see page 1236) ID for which to disable permission inheritance.

Returns

False if it's successful.

Permissions.EnableContentInheritance Method

Deletes current permissions from the content and enable permissions inheritance on the content.

This method can be accessed through the Permissions.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Permissions.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "EnableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool EnableContentInheritance(long id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "EnableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")>
Public Function EnableContentInheritance(id As long) As bool
```

Parameters

Parameters	Description
long id	The content's ID.

Returns

False if it's successful.

Permissions.EnableFolderInheritance Method

Deletes current permissions from the folder and enable permissions inheritance on the folder.

This method can be accessed through the Permissions.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Permissions.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "EnableFolderInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool EnableFolderInheritance(long id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "EnableFolderInheritance", Description = "Deletes a user or a group
from the item's permissions")>
Public Function EnableFolderInheritance(id As long) As bool
```

Parameters

Parameters	Description
long id	The Folder (see page 1236) ID.

Returns

False if it's successful.

GetUserPermissions Method

This is the overview for the GetUserPermissions method overload.

Description

This method is overloaded.

Overload List

	Name	Description
	Permissions.GetUserPermissions (long, string, long, string) (see page 1309)	Loads the user permissions as a array of UserPermissionData. This method can be accessed through the Permissions.asmx file located in:
	Permissions.GetUserPermissions (long, string, long, string, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) (see page 1309)	Loads the user permissions as a array of UserPermissionData. This method can be accessed through the Permissions.asmx file located in:

Permissions.GetUserPermissions Method (long, string, long, string)

Loads the user permissions as a array of UserPermissionData.

This method can be accessed through the Permissions.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Permissions.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserPermissions", Description = "Loads the user permissions as a
array of UserPermissionData.")]
public UserPermissionData[] GetUserPermissions(long Id, string ItemType, long UserId, string
UserGroupList);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserPermissions", Description = "Loads the user permissions as a
array of UserPermissionData.")>
Public Function GetUserPermissions(Id As long, ItemType As string, UserId As long,
UserGroupList As string) As UserPermissionData()
```

Parameters

Parameters	Description
long Id	The item's ID.
string ItemType	The item's type.
long UserId	The user's ID.
string UserGroupList	The group list.

Returns

Ektron.Cms.UserPermissionData()

Permissions.GetUserPermissions Method (long, string, long, string, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType)

Loads the user permissions as a array of UserPermissionData.

This method can be accessed through the Permissions.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Permissions.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserPermissions By PermissionType", Description = "Loads the user
permissions as a array of UserPermissionData.")]
public UserPermissionData[] GetUserPermissions(long Id, string ItemType, long UserId, string
UserGroupList, ContentAPI.PermissionUserType PermissionType, ContentAPI.PermissionRequestType
PermissionRequest);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserPermissions By PermissionType", Description = "Loads the user
permissions as a array of UserPermissionData.")>
Public Function GetUserPermissions(Id As long, ItemType As string, UserId As long,
UserGroupList As string, PermissionType As ContentAPI.PermissionUserType, PermissionRequest As
ContentAPI.PermissionRequestType) As UserPermissionData()
```

Parameters

Parameters	Description
long Id	The item's ID.
string ItemType	The item's type.
long UserId	The user's ID.
string UserGroupList	The group list.
ContentAPI.PermissionUserType PermissionType	PermissionUserType
ContentAPI.PermissionRequestType PermissionRequest	PermissionRequestType

Returns

Ektron.Cms.UserPermissionData()

Permissions.LoadPermissions Method

Loads the permission for given object, such as content, folder, task, etc.

This method can be accessed through the Permissions.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Permissions.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "LoadPermissions", Description = "Loads the permission for given
object, such as content, folder, task, etc.")]
public PermissionData LoadPermissions(long Id, string RequestType,
ContentAPI.PermissionResultType Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "LoadPermissions", Description = "Loads the permission for given
object, such as content, folder, task, etc.")>
Public Function LoadPermissions(Id As long, RequestType As string, Type As
ContentAPI.PermissionResultType) As PermissionData
```

Parameters

Parameters	Description
long Id	The object's ID.
string RequestType	The object type:= content | folder | collections | tasks
ContentAPI.PermissionResultType Type	The permission result type.

Returns

Ektron.Cms.PermissionData

Permissions.UpdateItemPermission Method

Updates the item permissions for a user or a usergroup.

This method can be accessed through the Permissions.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Permissions.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateItemPermission", Description = "Updates the item permissions
for a user or a usergroup.")]
```

```
public bool UpdateItemPermission(UserPermissionData userPermData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateItemPermission", Description = "Updates the item permissions
for a user or a usergroup.")>
Public Function UpdateItemPermission(userPermData As UserPermissionData) As bool
```

Parameters

Parameters	Description
UserPermissionData userPermData	User (see page 1369)'s Permissions (see page 1304) Data.

Returns

False if it's successful.

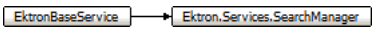
SearchManager Class

This section describes the Web Service methods used with the SearchManager. The SearchManager Web service is a wrapper for the SearchManager API (see page 669): `Ektron.Cms.API.Search.SearchManager` (see page 899). These methods can be accessed through the SearchManager.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Search/SearchManager.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class SearchManager : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class SearchManager
Inherits EktronBaseService
```

File

SearchManager.cs

SearchManager Methods

The methods of the SearchManager class are listed here.

Public Methods

	Name	Description
	ExecuteSearch (see page 1312)	This is the overview for the ExecuteSearch method overload.
	Search (see page 1313)	This is the overview for the Search method overload.
	SearchForCategories (see page 1315)	Search (see page 1313) based on taxonomy category. This method can be accessed through the SearchManager.asmx file located in:
	SearchTable (see page 1315)	Search (see page 1313) html, content, images, multimedia, documents based on content or metadata. This method can be accessed through the SearchManager.asmx file located in:

ExecuteSearch Method

This is the overview for the ExecuteSearch method overload.

Description

Returns the array of ContentData. This method is overloaded.

ExecuteSearch (ContentSearchCondition) (see page 1312)

ExecuteSearch (UserSearchCondition) (see page 1312)

Overload List

	Name	Description
	SearchManager.ExecuteSearch (ContentSearchCondition) (see page 1312)	Returns the array of ContentData based on the criteria ContentSearchCondition. This method can be accessed through the SearchManager.asmx file located in:
	SearchManager.ExecuteSearch (UserSearchCondition) (see page 1312)	Returns the array of UserData based on the criteria UserSearchCondition. This method can be accessed through the SearchManager.asmx file located in:

SearchManager.ExecuteSearch Method (ContentSearchCondition)

Returns the array of ContentData based on the criteria ContentSearchCondition.

This method can be accessed through the SearchManager.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Search/SearchManager.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ExecuteSearch For Content", Description = "")]
public ContentData[] ExecuteSearch(ContentSearchCondition condition);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ExecuteSearch For Content", Description = "")>
Public Function ExecuteSearch(condition As ContentSearchCondition) As ContentData()
```

Parameters

Parameters	Description
ContentSearchCondition condition	ContentSearchCondition

Returns

Array of ContentData

SearchManager.ExecuteSearch Method (UserSearchCondition)

Returns the array of UserData based on the criteria UserSearchCondition.

This method can be accessed through the SearchManager.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Search/SearchManager.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "ExecuteSearch For User", Description = "")]
public UserData[] ExecuteSearch(UserSearchCondition condition);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ExecuteSearch For User", Description = "")>
Public Function ExecuteSearch(condition As UserSearchCondition) As UserData()
```

Parameters

Parameters	Description
UserSearchCondition condition	UserSearchCondition

Returns

Array of UserData

Search Method

This is the overview for the Search method overload.

Description

This method is overloaded.

Overload List

	Name	Description
	SearchManager.Search (MapRequestData, int, int, int) (see page 1313)	Map (see page 350) search for retrieving geographical content from CMS400.Net site. This method can be accessed through the SearchManager.asmx file located in:
	SearchManager.Search (SearchRequestData, int) (see page 1314)	Search html, content, images, multimedia, documents based on content or metadata. This method can be accessed through the SearchManager.asmx file located in:
	SearchManager.Search (string, int) (see page 1314)	Search the site for occurrence(s) of search text. This method can be accessed through the SearchManager.asmx file located in:

SearchManager.Search Method (MapRequestData, int, int, int)

Map (see page 350) search for retrieving geographical content from CMS400.Net site.

This method can be accessed through the SearchManager.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Search/SearchManager.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ExecuteSearch search complete", Description = "")]
public MapResponseData[] Search(MapRequestData mapReqData, int index, int count, ref int
resultCount);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ExecuteSearch search complete", Description = "")>
Public Function Search(mapReqData As MapRequestData, index As Integer, count As Integer, ByRef
resultCount As Integer) As MapResponseData()
```

Parameters

Parameters	Description
MapRequestData mapReqData	Search criteria bundled in a MapRequestData object.
int index	Page number.
int count	Page size.
ref int resultCount	Number of records returned.
context	The current Http request.

Returns

Array of MapResponseData - content satisfying location criteria

SearchManager.Search Method (SearchRequestData, int)

Search html, content, images, multimedia, documents based on content or metadata.

This method can be accessed through the SearchManager.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Search/SearchManager.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ExecuteSearch search complete Search Request Data", Description =
"")]
public SearchResponseData[] Search(SearchRequestData searchReqData, ref int resultCount);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ExecuteSearch search complete Search Request Data", Description =
"")>
Public Function Search(searchReqData As SearchRequestData, ByRef resultCount As Integer) As
SearchResponseData()
```

Parameters

Parameters	Description
SearchRequestData searchReqData	Search criteria bundled in a SearchRequestData object.
ref int resultCount	Number of records returned.
context	The current Http request.

Returns

Array of SearchResponseData - content satisfying search criteria

SearchManager.Search Method (string, int)

Search the site for occurrence(s) of search text.

This method can be accessed through the SearchManager.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Search/SearchManager.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ExecuteSearch basic", Description = "")]
public SearchResponseData[] Search(string searchText, ref int resultCount);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ExecuteSearch basic", Description = "")>
Public Function Search(searchText As string, ByRef resultCount As Integer) As
SearchResponseData()
```

Parameters

Parameters	Description
ref int resultCount	Number of records returned.
searchReqData	Search criteria bundled in a SearchRequestData object.

SearchManager.SearchForCategories Method

Search (see page 1313) based on taxonomy category.

This method can be accessed through the SearchManager.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Search/SearchManager.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "Search for categories", Description = "")]
public String[] SearchForCategories(SearchRequestData searchReqData, int index, int count, ref
int resultCount);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "Search for categories", Description = "")>
Public Function SearchForCategories(searchReqData As SearchRequestData, index As Integer,
count As Integer, ByRef resultCount As Integer) As String()
```

Parameters

Parameters	Description
SearchRequestData searchReqData	Search (see page 1313) criteria bundled in a SearchRequestData object.
int index	Page number.
int count	Page size.
ref int resultCount	Number of records returned.
context	The current Http request.

Returns

Array of String - List of categories

SearchManager.SearchTable Method

Search (see page 1313) html, content, images, multimedia, documents based on content or metadata.

This method can be accessed through the SearchManager.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Search/SearchManager.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "Return SearchData Table", Description = "")]
public DataSet SearchTable(SearchRequestData searchReqData, string queryParams, ref int
resultCount);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "Return SearchData Table", Description = "")>
Public Function SearchTable(searchReqData As SearchRequestData, queryParams As string, ByRef
resultCount As Integer) As DataSet
```

Parameters

Parameters	Description
SearchRequestData searchReqData	Search (see page 1313) criteria bundled in a SearchRequestData object.
ref int resultCount	Number of records returned.
context	The current Http request.

Returns

Array of SearchResponseData - content satisfying search criteria

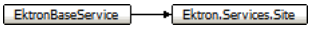
Site Class

This section describes the Web Service methods used with Site functionality. The Site Web service is a wrapper for the Site API (see page 669): `Ektron.Cms.API.Site (see page 725)`. These methods can be accessed through the Site.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Site.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Site : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Site
Inherits EktronBaseService
```

File

Site.cs

Site Methods

The methods of the Site class are listed here.

Public Methods

	Name	Description
	BreakTaskPermissionInherit (see page 1317)	Removes the task permissions for a given user. This method can be accessed through the Site.asmx file located in:
	GetAllActiveLanguages (see page 1317)	Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language Settings area. This method can be accessed through the Site.asmx file located in:
	GetAllLanguages (see page 1318)	This is the overview for the GetAllLanguages method overload.
	GetClientDayNames (see page 1319)	Returns the day name of a given content language (or site language). This method can be accessed through the Site.asmx file located in:

	GetClientMonthNames (see page 1319)	Returns the month name of a given content language (or site language) This method can be accessed through the Site.asmx file located in:
	GetCookieValue (see page 1320)	Reads the cookie value. This method can be accessed through the Site.asmx file located in:
	GetLanguage (see page 1320)	Returns the details about the given language ID. This method can be accessed through the Site.asmx file located in:
	GetLanguages (see page 1321)	Gets a list of all languages that meet a given state, for example, active. This method can be accessed through the Site.asmx file located in:
	GetSiteVariables (see page 1321)	Returns all of the settings data. This method can be accessed through the Site.asmx file located in:
	GetTaskPermission (see page 1322)	Use this method to find out whether a user has task permissions. This method can be accessed through the Site.asmx file located in:
	IsTaskPermissionInherited (see page 1322)	Returns true if a user has already granted one of the task permissions. This method can be accessed through the Site.asmx file located in:
	SetCookieValue (see page 1323)	Resets the cookie value. This method can be accessed through the Site.asmx file located in:
	SetTaskPermission (see page 1323)	Updates the task permission(s) for the given user or group. This method can be accessed through the Site.asmx file located in:
	UpdateSiteEnabledLanguages (see page 1324)	Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled. This method can be accessed through the Site.asmx file located in:

Site.BreakTaskPermissionInherit Method

Removes the task permissions for a given user.

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "BreakTaskPermissionInherit", Description = "")]
public void BreakTaskPermissionInherit(long UserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "BreakTaskPermissionInherit", Description = "")>
Public Function BreakTaskPermissionInherit(UserId As long) As void
```

Parameters

Parameters	Description
long UserId	User (see page 1369) ID

Remarks

The user should be logged in to the system. Otherwise, the user will get the permission denied error.

Site.GetAllActiveLanguages Method

Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language Settings area.

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllActiveLanguages", Description = "")]
public LanguageData[] GetAllActiveLanguages();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllActiveLanguages", Description = "")>
Public Function GetAllActiveLanguages() As LanguageData()
```

Returns

Array of Ektron.Cms.LanguageData

GetAllLanguages Method

This is the overview for the GetAllLanguages method overload.

Description

This method is overloaded.

- GetAllLanguages () (see page 1318) - gets a list of all the available languages sorted by language name in ascending order.
- GetAllLanguages (LanguageDataSet.SortBy, LanguageDataSet.SortDirection) (see page 1318) - gets a list of all the available languages and allows you to choose the order direction of the list.

Overload List

	Name	Description
	Site.GetAllLanguages () (see page 1318)	Gets a list of all the available languages sorted by language name in ascending order. This method can be accessed through the Site.asmx file located in:
	Site.GetAllLanguages (LanguageDataSet.SortBy, LanguageDataSet.SortDirection) (see page 1318)	Gets a list of all the available languages. This method can be accessed through the Site.asmx file located in:

Site.GetAllLanguages Method ()

Gets a list of all the available languages sorted by language name in ascending order.

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllLanguagesSortByName", Description = "")]
public LanguageData[] GetAllLanguages();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllLanguagesSortByName", Description = "")>
Public Function GetAllLanguages() As LanguageData()
```

Returns

Array of Ektron.Cms.LanguageData sorted by

Site.GetAllLanguages Method (LanguageDataSet.SortBy, LanguageDataSet.SortDirection)

Gets a list of all the available languages.

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllLanguages", Description = "")]
public LanguageData[] GetAllLanguages(LanguageDataSet.SortBy orderBy,
LanguageDataSet.SortDirection direction);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllLanguages", Description = "")>
Public Function GetAllLanguages(orderBy As LanguageDataSet.SortBy, direction As
LanguageDataSet.SortDirection) As LanguageData()
```

Parameters

Parameters	Description
LanguageDataSet.SortBy orderBy	Determines the sort order.
LanguageDataSet.SortDirection direction	Determines the sort direction.

Returns

Array of Ektron.Cms.LanguageData

Site.GetClientDayNames Method

Returns the day name of a given content language (or site language).

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetClientDayNames", Description = "")]
public string GetClientDayNames();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetClientDayNames", Description = "")>
Public Function GetClientDayNames() As string
```

Returns

String

Site.GetClientMonthNames Method

Returns the month name of a given content language (or site language)

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetClientMonthNames", Description = "")]
public string GetClientMonthNames();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetClientMonthNames", Description = "")>
Public Function GetClientMonthNames() As string
```

Returns

String

Site.GetCookieValue Method

Reads the cookie value.

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCookieValue", Description = "")]
public string GetCookieValue(string Keyname);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCookieValue", Description = "")>
Public Function GetCookieValue(Keyname As string) As string
```

Parameters

Parameters	Description
string Keyname	The Keyname. (Required)

Returns

String

Site.GetLanguage Method

Returns the details about the given language ID.

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLanguage", Description = "")]
public LanguageData GetLanguage(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLanguage", Description = "")>
Public Function GetLanguage(Id As Integer) As LanguageData
```

Parameters

Parameters	Description
int Id	Valid non negative language ID (refer to the language_type table for all the languages)

Returns

Ektron.Cms.LanguageData

Site.GetLanguages Method

Gets a list of all languages that meet a given state, for example, active.

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLanguages", Description = "")]
public LanguageDataSet.LanguageDataTable GetLanguages(LanguageDataSet.LanguageState State,
LanguageDataSet.SortBy SortBy, LanguageDataSet.SortDirection SortDirection);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLanguages", Description = "")>
Public Function GetLanguages(State As LanguageDataSet.LanguageState, SortBy As
LanguageDataSet.SortBy, SortDirection As LanguageDataSet.SortDirection) As
LanguageDataSet.LanguageDataTable
```

Parameters

Parameters	Description
LanguageDataSet.LanguageState State	The state that the language must meet. For example, defined, active, site enabled.
LanguageDataSet.SortBy SortBy	The field on which to sort the languages in the data table.
LanguageDataSet.SortDirection SortDirection	The direction, ascending or descending, in which to sort the field values.

Returns

LanguageDataTable of languages that meet the given state.

Site.GetSiteVariables Method

Returns all of the settings data.

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetSiteVariables", Description = "")]
public SettingsData GetSiteVariables(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetSiteVariables", Description = "")>
Public Function GetSiteVariables(Id As long) As SettingsData
```

Parameters

Parameters	Description
long Id	User (see page 1369) ID

Returns

Ektron.Cms.SettingsData

Remarks

If user ID is passed it returns the additional value of MaxUsersAllowed, AdInstalled and AdAllowed properties.

Site.GetTaskPermission Method

Use this method to find out whether a user has task permissions.

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaskPermission", Description = "")]
public long GetTaskPermission(long Id, int Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTaskPermission", Description = "")>
Public Function GetTaskPermission(Id As long, Type As Integer) As long
```

Parameters

Parameters	Description
long Id	User (see page 1369) or group ID.
int Type	User (see page 1369) or group (1 = user, 2 = group).

Returns

Returns 1 if the user has one of the task permissions, otherwise 0.

Remarks

The user should be logged in to the system. Otherwise, the user will get the permission denied error.

Site.IsTaskPermissionInherited Method

Returns true if a user has already granted one of the task permissions.

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsTaskPermissionInherited", Description = "")]
public bool IsTaskPermissionInherited(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsTaskPermissionInherited", Description = "")>
Public Function IsTaskPermissionInherited(Id As long) As bool
```

Parameters

Parameters	Description
long Id	User (see page 1369) ID.

Returns

Boolean

Remarks

To use this function, the user should be logged in to the system. Otherwise, the user will get the permission denied error.

Site.SetCookieValue Method

Resets the cookie value.

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SetCookieValue", Description = "")]
public void SetCookieValue(string Keyname, string CookieValue);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SetCookieValue", Description = "")>
Public Function SetCookieValue(Keyname As string, CookieValue As string) As void
```

Parameters

Parameters	Description
string Keyname	The Keyname. (Required)
string CookieValue	Value

Site.SetTaskPermission Method

Updates the task permission(s) for the given user or group.

This method can be accessed through the Site.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Site.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SetTaskPermission", Description = "")]
public void SetTaskPermission(long Id, int UserType, int PermissionType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SetTaskPermission", Description = "")>
Public Function SetTaskPermission(Id As long, UserType As Integer, PermissionType As Integer)
As void
```

Parameters

Parameters	Description
long Id	User (see page 1369) ID or Group ID
int UserType	User (see page 1369) (1) or Group (2)

int PermissionType	Ektron.Cms.Common.TaskPermission (1 = Create Task (see page 1085), 2 = Redirect Task (see page 1085), 3 = Delete Task (see page 1085), 4 = Edit Alias)

Remarks

To use this function, the user should be logged in to the system. Otherwise, user will get the permission denied error.

Site.UpdateSiteEnabledLanguages Method

Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled.

This method can be accessed through the Site.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Site.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateSiteEnabledLanguages", Description = "")]
public void UpdateSiteEnabledLanguages(string LanguageIDs);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateSiteEnabledLanguages", Description = "")>
Public Function UpdateSiteEnabledLanguages(LanguageIDs As string) As void
```

Parameters

Parameters	Description
string LanguageIDs	Comma-separated list of language IDs. For example, "1033,1031"

SiteMap Class

This section describes the Web Service methods used with SiteMap. The SiteMap Web service is a wrapper for the SiteMap API (see page 669): `Ektron.Cms.API.SiteMap` (see page 732). These methods can be accessed through the SiteMap.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/SiteMap.asmx`

Web Services (see page 1083) examples involving content are available in the Developer sample site. You can download this site from:

http://www.ektron.com/cms400-web-cms.aspx?id=5278

Once the developer site is installed, example code files are located in your site root. For example:

`C:\Inetpub\wwwroot\CMS400Developer\Developer\dotNetAssemblies\WebServicesAPI\SiteAPI`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header. For example:

```
sitemapApi.AuthenticationHeaderValue = New localhost.AuthenticationHeader()
sitemapApi.AuthenticationHeaderValue.Username = "admin"
sitemapApi.AuthenticationHeaderValue.Password = "admin"
sitemapApi.AuthenticationHeaderValue.Domain = "Domain" '''Only needed if AD/LDAP is being used
```

Class Hierarchy

C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
```

```
public class SiteMap : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class SiteMap
Inherits EktronBaseService
```

File

SiteMap.cs

SiteMap Methods

The methods of the SiteMap class are listed here.

Public Methods

	Name	Description
	GetFolderBreadcrumbPath (see page 1325)	Get the breadcrumb path of the folder in which content block exists. This method can be accessed through the SiteMap.asmx file located in:
	GetWebsiteMap (see page 1325)	Creates the site map from folder breadcrumb. This method can be accessed through the SiteMap.asmx file located in:
	RenderFolderBreadcrumbHtml (see page 1326)	Render folder breadcrumb HTML. This method can be accessed through the SiteMap.asmx file located in:
	RenderWebsiteMap (see page 1327)	Create HTML to render on the page. This method can be accessed through the SiteMap.asmx file located in:

SiteMap.GetFolderBreadcrumbPath Method

Get the breadcrumb path of the folder in which content block exists.

This method can be accessed through the SiteMap.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/SiteMap.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolderBreadcrumbPath", Description = "")]
public SitemapPath[] GetFolderBreadcrumbPath(long id, bool isFolder);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolderBreadcrumbPath", Description = "")>
Public Function GetFolderBreadcrumbPath(id As long, isFolder As bool) As SitemapPath()
```

Parameters

Parameters	Description
long id	A Content (see page 1198) or Folder (see page 1236) ID.
bool isFolder	Set to True if the ID is folder ID.

Returns

Array of nodes in the path.

SiteMap.GetWebsiteMap Method

Creates the site map from folder breadcrumb.

This method can be accessed through the SiteMap.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/SiteMap.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetWebsiteMap", Description = "")]
public WebsiteMap GetWebsiteMap(long folderId, bool getSameLevelNodes);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetWebsiteMap", Description = "")>
Public Function GetWebsiteMap(folderId As long, getSameLevelNodes As bool) As WebsiteMap
```

Parameters

Parameters	Description
long folderId	Build the Web sitemap from the specified folder.
bool getSameLevelNodes	Set to True to include all sites at same level as the folder ID.

SiteMap.RenderFolderBreadcrumbHtml Method

Render folder breadcrumb HTML.

This method can be accessed through the SiteMap.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/SiteMap.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RenderFolderBreadcrumbHtml", Description = "")]
public string RenderFolderBreadcrumbHtml(SitemapPath[] nodes, string pathSeparator, bool
linkNodes, string linkTarget, bool displayVertical, string pageUrl);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RenderFolderBreadcrumbHtml", Description = "")>
Public Function RenderFolderBreadcrumbHtml(nodes As SitemapPath[], pathSeparator As string,
linkNodes As bool, linkTarget As string, displayVertical As bool, pageUrl As string) As string
```

Parameters

Parameters	Description
SitemapPath[] nodes	The items contained in the breadcrumb path. Ektron.Cms.Common.SitemapPath
string pathSeparator	A special character that separates the items in a breadcrumb trail.
bool linkNodes	Should the items in the breadcrumb path be Hyperlinked? • **True** = hyperlink breadcrumb items. • **False** = breadcrumb items appear as plain text.
string linkTarget	Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self. • **_Self** - opens link in the same window. • **_Top** - opens link in a parent window. • **_Blank** - opens link in a new window. • **_Parent** - opens link in the parent frame.

displayVerticle	Set to True to display the breadcrumb vertically. Set to false to display horizontally. • **True** = display the breadcrumb trail vertically. • **False** = display the breadcrumb trail horizontally.

Returns

HTML

Description

SiteMap.RenderWebsiteMap Method

Create HTML to render on the page.

This method can be accessed through the SiteMap.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/SiteMap.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RenderWebsiteMap", Description = "")]
public string RenderWebsiteMap(WebsiteMap webSitemap, string displayType, string className,
int maxLevel, int startLevel, string pageUrl);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RenderWebsiteMap", Description = "")>
Public Function RenderWebsiteMap(webSitemap As WebsiteMap, displayType As string, className As
string, maxLevel As Integer, startLevel As Integer, pageUrl As string) As string
```

Parameters

Parameters	Description
WebsiteMap webSitemap	GetWebSiteMap()
string displayType	Determines how to produce HTML.
string className	Stylesheet class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.
int maxLevel	Set the max level of the sitemap. Set it to 0 for unlimited.
int startLevel	Set the starting level of the sitemap. Set it to 0 for start from root.

Returns

HTML/XML

Remarks

Supported displayType: ListWithDescription | List | RawXml

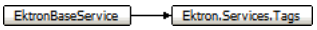
Tags Class

This section describes the Web Service methods used with Ektron's Tags feature. This feature is a part Ektron's Community (see page 750) Platform. The Tags Web service is a wrapper for the Tags API (see page 669): `Ektron.Cms.API.Community.Tags` (see page 805). These methods can be accessed through the Tags.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
public class Tags : EktronBaseService;
```

Visual Basic

```
Public Class Tags
Inherits EktronBaseService
```

File

Tags.cs

Description

[WebService(Namespace = "http://microsoft.com/webservices/")]

Tags Methods

The methods of the Tags class are listed here.

Public Methods

	Name	Description
	AddTag (see page 1329)	Saves a Tag and returns new ID. If the tag already exists, the existing Tag ID is returned. This method can be accessed through the Tags.asmx file located in:
	AddTagToCommunityGroup (see page 1329)	This is the overview for the AddTagToCommunityGroup method overload.
	AddTagToContent (see page 1332)	This is the overview for the AddTagToContent method overload.
	AddTagToObject (see page 1333)	This is the overview for the AddTagToObject method overload.
	AddTagToUser (see page 1335)	This is the overview for the AddTagToUser method overload.
	DeleteTagById (see page 1337)	Permanently deletes tag and all associations. Even deletes system tags so this should only be called if user has permission to delete system tags. If deleting tags used by a specific user, use DeleteTagsByUser (see page 1338). This method can be accessed through the Tags.asmx file located in:
	DeleteTagOnObject (see page 1338)	Deletes a user's tag on an object. For example, A user had tagged content ID #101 with "CMS" and now wishes to delete that object tag. This method can be accessed through the Tags.asmx file located in:
	DeleteTagsByUser (see page 1338)	Deletes all tags applied by a user. To delete tags ON a user, use DeleteTagsOnUser (see page 1340). This method can be accessed through the Tags.asmx file located in:
	DeleteTagsOnCommunityGroup (see page 1339)	Deletes all tags on a piece of content. This method can be accessed through the Tags.asmx file located in:
	DeleteTagsOnContent (see page 1339)	Deletes all tags on a piece of content. This method can be accessed through the Tags.asmx file located in:
	DeleteTagsOnObject (see page 1340)	Deletes all tags on an object. This method can be accessed through the Tags.asmx file located in:
	DeleteTagsOnUser (see page 1340)	Deletes all tags applied to a user. To delete tags applied BY a given user, use DeleteTagsByUser (see page 1338). This method can be accessed through the Tags.asmx file located in:
	EditTag (see page 1341)	Updates an existing Tag. This method can be accessed through the Tags.asmx file located in:
	GetAllTagRelatedContent (see page 1341)	Returns a list of content related to the supplied Content (see page 1198) via a common Tag. For example, both content item one and content item two are tagged with "CMS" and are considered related. This method can be accessed through the Tags.asmx file located in:
	GetAllTagRelatedUsers (see page 1342)	Returns a list of users related to the supplied user via a common Tag. For example, both user one and user two are both tagged with "Engineer" and are considered related. This method can be accessed through the Tags.asmx file located in:

	GetAllTags (see page 1342)	This is the overview for the GetAllTags method overload.
	GetDefaultTags (see page 1344)	This is the overview for the GetDefaultTags method overload.
	GetTagByID (see page 1345)	Returns the Tag with a specified ID. This method can be accessed through the Tags.asmx file located in:
	GetTagByName (see page 1346)	Returns Tag with specified Text. This method can be accessed through the Tags.asmx file located in:
	GetTagCloud (see page 1346)	Returns a TagCloud (see page 513) based upon the request. This method can be accessed through the Tags.asmx file located in:
	GetTagsForContent (see page 1347)	Returns an array of tags for a piece of content. This method can be accessed through the Tags.asmx file located in:
	GetTagsForObject (see page 1347)	This is the overview for the GetTagsForObject method overload.
	GetTagsForUser (see page 1348)	This is the overview for the GetTagsForUser method overload.
	RemoveTagAsDefault (see page 1350)	Removes a tag from the object's default tag list. This method can be accessed through the Tags.asmx file located in:
	SaveTagAsDefault (see page 1350)	Saves a tag to the the object's default list. This method can be accessed through the Tags.asmx file located in:

Tags.AddTag Method

Saves a Tag and returns new ID. If the tag already exists, the existing Tag ID is returned.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTag", Description = "Saves a Tag and returns new Id. If the tag
already exists, the existing Tag Id is returned.")]
public long AddTag(ref TagData tag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTag", Description = "Saves a Tag and returns new Id. If the tag
already exists, the existing Tag Id is returned.")>
Public Function AddTag(ByRef tag As TagData) As long
```

Parameters

Parameters	Description
ref TagData tag	Tag data to save.

Returns

An ID for the newly created tag or existing tag.

AddTagToCommunityGroup Method

This is the overview for the AddTagToCommunityGroup method overload.

Description

This method is overloaded.

- AddTagToCommunityGroup (String, Long) - applies a tag to a community group by providing the tag's text and the community group's ID.
- AddTagToCommunityGroup (String, Long, Int) - applies a tag in a specified language to a community group by providing the tag's text, the community group's ID, and the language ID.

- AddTagToCommunityGroup (String, Long, Long, Int) - applies a user created tag in a specified language to a community group by providing the tag's text, the community group's ID, the user ID and the language ID.

Overload List

	Name	Description
	Tags.AddTagToCommunityGroup (string, long) (see page 1330)	Applies a Tag to a community group. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language. This method can be accessed through the Tags.asmx file located in:
	Tags.AddTagToCommunityGroup (string, long, int) (see page 1331)	Applies a Tag to a community group in a specified language. If the tag text does not already exist, it is created. Uses the currently logged in user. The newly created ID is accessible via the returned TagData. This method can be accessed through the Tags.asmx file located in:
	Tags.AddTagToCommunityGroup (string, long, long, int) (see page 1331)	Applies a Tag to a community group in a specified language by a given language. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. This method can be accessed through the Tags.asmx file located in:

Tags.AddTagToCommunityGroup Method (string, long)

Applies a Tag to a community group. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language.

This method can be accessed through the Tags.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToCommunityGroup", Description = "Applies a Tag to a piece of
content.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData.  Adds the tag for the current user and the Tag is saved
under the current site language.")]
public TagData AddTagToCommunityGroup(string tagText, long groupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToCommunityGroup", Description = "Applies a Tag to a piece of
content.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData.  Adds the tag for the current user and the Tag is saved
under the current site language.")>
Public Function AddTagToCommunityGroup(tagText As string, groupId As long) As TagData
```

Parameters

Parameters	Description
string tagText	The Text of the tag being applied to the community group.
long groupId	The ID of the community group that is being tagged.

Returns

TagData structure representing the applied tag.

Tags.AddTagToCommunityGroup Method (string, long, int)

Applies a Tag to a community group in a specified language. If the tag text does not already exist, it is created. Uses the currently logged in user. The newly created ID is accessible via the returned TagData.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToCommunityGroup with language", Description = "Applies a Tag
to a piece of content.  If the tag text does not already exist, it is created.  Uses the
currently logged in user. The newly created Id is accessible via the returned TagData. ")]
public TagData AddTagToCommunityGroup(string tagText, long groupId, int languageId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToCommunityGroup with language", Description = "Applies a Tag
to a piece of content.  If the tag text does not already exist, it is created.  Uses the
currently logged in user. The newly created Id is accessible via the returned TagData. ")>
Public Function AddTagToCommunityGroup(tagText As string, groupId As long, languageId As
Integer) As TagData
```

Parameters

Parameters	Description
string tagText	Text of tag being applied to a community group.
long groupId	The ID of the community that is being tagged.
int languageId	The ID of the language the to apply to the tagText. Supply -1 for all languages.

Returns

TagData structure representing the applied tag.

Tags.AddTagToCommunityGroup Method (string, long, long, int)

Applies a Tag to a community group in a specified language by a given language. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToCommunityGroup with user and language", Description =
"Applies a Tag to a piece of content.  If the tag text does not already exist, it is created.
The newly created Id is accessible via the returned TagData.")]
public TagData AddTagToCommunityGroup(string tagText, long groupId, long userId, int
languageId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToCommunityGroup with user and language", Description =
"Applies a Tag to a piece of content.  If the tag text does not already exist, it is created.
The newly created Id is accessible via the returned TagData.")>
Public Function AddTagToCommunityGroup(tagText As string, groupId As long, userId As long,
languageId As Integer) As TagData
```

Parameters

Parameters	Description
string tagText	The Text of the tag being applied to a community group.
long groupId	The ID of the community group that is being tagged.
long userId	The ID of the user applying the tag.
int languageId	The ID of the language being applied to the tagText. Supply -1 for all languages.

Returns

TagData structure representing the applied tag.

AddTagToContent Method

This is the overview for the AddTagToContent method overload.

Description

This method is overloaded.

- AddTagToContent (String, Long) - applies a tag to a piece of content based on the tag's text and the content's ID.
- AddTagToContent (String, Long, Long, Int) - applies a user created tag in a specified language to a content item by providing the tag's text, the content's ID, the user ID and the language ID.

Overload List

	Name	Description
	Tags.AddTagToContent (string, long) (see page 1332)	Applies a Tag to a content item. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language. This method can be accessed through the Tags.asmx file located in:
	Tags.AddTagToContent (string, long, long, int) (see page 1333)	Applies a Tag to a content item for a specified user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. This method can be accessed through the Tags.asmx file located in:

Tags.AddTagToContent Method (string, long)

Applies a Tag to a content item. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToContent", Description = "Applies a Tag to a piece of
content.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData. Adds the tag for the current user and the Tag is saved
under the current site language.")]
public TagData AddTagToContent(string tagText, long contentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToContent", Description = "Applies a Tag to a piece of
content.  If the tag text does not already exist, it is created.  The newly created Id is
```

```
accessible via the returned TagData. Adds the tag for the current user and the Tag is saved
under the current site language.")>
Public Function AddTagToContent(tagText As string, contentId As long) As TagData
```

Parameters

Parameters	Description
string tagText	Text of the tag to apply to object.
long contentId	The ID of the content item that is being tagged.

Returns

TagData structure representing the applied tag.

Tags.AddTagToContent Method (string, long, long, int)

Applies a Tag to a content item for a specified user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToContent with user and language", Description = "Applies a
Tag to a piece of content.  If the tag text does not already exist, it is created.  The newly
created Id is accessible via the returned TagData.")]
public TagData AddTagToContent(string tagText, long contentId, long userId, int languageId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToContent with user and language", Description = "Applies a
Tag to a piece of content.  If the tag text does not already exist, it is created.  The newly
created Id is accessible via the returned TagData.")>
Public Function AddTagToContent(tagText As string, contentId As long, userId As long,
languageId As Integer) As TagData
```

Parameters

Parameters	Description
string tagText	Text of tag to apply to object.
long contentId	The ID of content item that is being tagged.
long userId	The ID of the user applying the tag.
int languageId	The ID of the language to apply to the tagText. Supply -1 for all languages.

Returns

TagData structure representing the applied tag.

AddTagToObject Method

This is the overview for the AddTagToObject method overload.

Description

This method is overloaded.

- AddTagToObject (String, Long, EkEnumeration.CMSObjectTypes) - applies a tag to an object based on the tag's text and the object's type and ID.
- AddTagToObject (String, Int, EkEnumeraiton.CMSObjectType, Long, Int) - applies a user created tag in a specified language to an object by providing the tag's text, the object's type and ID, and the language ID.

Overload List

	Name	Description
	Tags.AddTagToObject (string, long, EkEnumeration.CMSObjectTypes) (see page 1334)	Applies a Tag to an object. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language. This method can be accessed through the Tags.asmx file located in:
	Tags.AddTagToObject (string, long, EkEnumeration.CMSObjectTypes, long, int) (see page 1334)	Applies a Tag to an object by a given user in a specified language. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. This method can be accessed through the Tags.asmx file located in:

Tags.AddTagToObject Method (string, long, EkEnumeration.CMSObjectTypes)

Applies a Tag to an object. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToObject", Description = "Applies a Tag to a piece of
content.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData.  Adds the tag for the current user and the Tag is saved
under the current site language.")]
public TagData AddTagToObject(string tagText, long objectId, EkEnumeration.CMSObjectTypes
objectType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToObject", Description = "Applies a Tag to a piece of
content.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData.  Adds the tag for the current user and the Tag is saved
under the current site language.")>
Public Function AddTagToObject(tagText As string, objectId As long, objectType As
EkEnumeration.CMSObjectTypes) As TagData
```

Parameters

Parameters	Description
string tagText	The text of a tag to being applied to an object.
long objectId	The ID of the object being tagged.
EkEnumeration.CMSObjectTypes objectType	The type of object represented by objectId.

Returns

TagData structure representing the applied tag.

Tags.AddTagToObject Method (string, long, EkEnumeration.CMSObjectTypes, long, int)

Applies a Tag to an object by a given user in a specified language. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToObject with user and language", Description = "Applies a Tag
to a piece of content.  If the tag text does not already exist, it is created.  The newly
created Id is accessible via the returned TagData.")]
public TagData AddTagToObject(string tagText, long objectId, EkEnumeration.CMSObjectTypes
objectType, long userId, int languageId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToObject with user and language", Description = "Applies a Tag
to a piece of content.  If the tag text does not already exist, it is created.  The newly
created Id is accessible via the returned TagData.")>
Public Function AddTagToObject(tagText As string, objectId As long, objectType As
EkEnumeration.CMSObjectTypes, userId As long, languageId As Integer) As TagData
```

Parameters

Parameters	Description
string tagText	The text of the tag being applied to an object.
long objectId	The ID of the object that is being tagged.
EkEnumeration.CMSObjectTypes objectType	The type of object represented by objectId.
long userId	The ID of the user applying the tag.
int languageId	The ID of the language being applied to the tagText. Supply -1 for all languages.

Returns

TagData structure representing the applied tag.

AddTagToUser Method

This is the overview for the AddTagToUser method overload.

Description

This method is overloaded.

- AddTagToUser (Long, Long) - applies a tag to a user based on a tag's ID and the user's ID. Use this method when a tag already exists.
- AddTagToUser (String, Long) - applies a tag to a user based on the tag's text and the user's ID. Use this method when you are creating a new tag for a user.
- AddTagToUser (String, Long, Int) - applies a tag in a specified language to a user by providing the tag's text, the user ID and the language ID.

Overload List

	Name	Description
	Tags.AddTagToUser (long, long) (see page 1336)	Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. This method can be accessed through the Tags.asmx file located in:
	Tags.AddTagToUser (string, long) (see page 1336)	Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. The Tag is saved under the current site language. This method can be accessed through the Tags.asmx file located in:

	Tags.AddTagToUser (string, long, int) (see page 1337)	Applies a Tag to a user for a specified language. If the tag text does not already exist, it is created. The newly created Id is accessible via the returned TagData. This method can be accessed through the Tags.asmx file located in:

Tags.AddTagToUser Method (long, long)

Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToUser with tag id", Description = "Applies a Tag to a user.
If the tag text does not already exist, it is created.  The newly created Id is accessible via
the returned TagData.")]
public void AddTagToUser(long tagId, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToUser with tag id", Description = "Applies a Tag to a user.
If the tag text does not already exist, it is created.  The newly created Id is accessible via
the returned TagData.")>
Public Function AddTagToUser(tagId As long, userId As long) As void
```

Parameters

Parameters	Description
long tagId	The ID of tag to apply to object.
long userId	The ID of the user that is being tagged.

Tags.AddTagToUser Method (string, long)

Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. The Tag is saved under the current site language.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToUser", Description = "Applies a Tag to a user.  If the tag
text does not already exist, it is created.  The newly created Id is accessible via the
returned TagData. The Tag is saved under the current site language.")]
public TagData AddTagToUser(string tagText, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToUser", Description = "Applies a Tag to a user.  If the tag
text does not already exist, it is created.  The newly created Id is accessible via the
returned TagData. The Tag is saved under the current site language.")>
Public Function AddTagToUser(tagText As string, userId As long) As TagData
```

Parameters

Parameters	Description
string tagText	The text of the tag being applied to the user.
long userId	The ID of the user that is being tagged.

Returns

TagData structure representing the applied tag.

Tags.AddTagToUser Method (string, long, int)

Applies a Tag to a user for a specified language. If the tag text does not already exist, it is created. The newly created Id is accessible via the returned TagData.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToUser with language", Description = "Applies a Tag to a
user.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData.")]
public TagData AddTagToUser(string tagText, long userId, int languageId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToUser with language", Description = "Applies a Tag to a
user.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData.")>
Public Function AddTagToUser(tagText As string, userId As long, languageId As Integer) As
TagData
```

Parameters

Parameters	Description
string tagText	The text of the tag being applied to the user.
long userId	The ID of the user that is being tagged.
int languageId	The ID of the language being applied to the tagText.

Returns

TagData structure representing the applied tag.

Tags.DeleteTagById Method

Permanently deletes tag and all associations. Even deletes system tags so this should only be called if user has permission to delete system tags. If deleting tags used by a specific user, use DeleteTagsByUser (see page 1338).

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagById", Description = "Permanently deletes tag and all
associations.  Even deletes system tags so this should only be called if user has permission
to delete system tags. If deleting tags used by a specific user, use DeleteTagsByUser.")]
public bool DeleteTagById(long tagId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagById", Description = "Permanently deletes tag and all
associations.  Even deletes system tags so this should only be called if user has permission
to delete system tags. If deleting tags used by a specific user, use DeleteTagsByUser.")>
Public Function DeleteTagById(tagId As long) As bool
```

Parameters

Parameters	Description
long tagId	ID of tag to be deleted.

Tags.DeleteTagOnObject Method

Deletes a user's tag on an object. For example, A user had tagged content ID #101 with "CMS" and now wishes to delete that object tag.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagOnObject", Description = "Deletes a user's tag on an
object. For example, User A had tagged content 101 with \"CMS\" and now wishes to delete that
object tag.")]
public bool DeleteTagOnObject(long tagID, long ObjectID, EkEnumeration.CMSObjectTypes
ObjectType, long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagOnObject", Description = "Deletes a user's tag on an
object. For example, User A had tagged content 101 with \"CMS\" and now wishes to delete that
object tag.")>
Public Function DeleteTagOnObject(tagID As long, ObjectID As long, ObjectType As
EkEnumeration.CMSObjectTypes, userId As long) As bool
```

Parameters

Parameters	Description
long tagID	The ID of tag to delete
long ObjectID	The ID of the object.
EkEnumeration.CMSObjectTypes ObjectType	The type of object associated with the ObjectId.
long userId	The ID of the user who originally applied the tag and wants to delete it.

Returns

True if operation was successful.

Tags.DeleteTagsByUser Method

Deletes all tags applied by a user. To delete tags ON a user, use DeleteTagsOnUser (see page 1340).

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagsByUser", Description = "Deletes all tags applied by a
```

```
user.")]
public bool DeleteTagsByUser(long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagsByUser", Description = "Deletes all tags applied by a
user.")>
Public Function DeleteTagsByUser(userId As long) As bool
```

Parameters

Parameters	Description
long userId	The ID of the user whose tags are being deleted

Returns

True if operation was successful.

Tags.DeleteTagsOnCommunityGroup Method

Deletes all tags on a piece of content.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagsOnCommunityGroup", Description = "Deletes all tags on a
piece of content.")]
public bool DeleteTagsOnCommunityGroup(long groupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagsOnCommunityGroup", Description = "Deletes all tags on a
piece of content.")>
Public Function DeleteTagsOnCommunityGroup(groupId As long) As bool
```

Parameters

Parameters	Description
long groupId	The ID of the community group whose tags are being deleted.

Returns

True if operation was successful

Tags.DeleteTagsOnContent Method

Deletes all tags on a piece of content.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagsOnContent", Description = "Deletes all tags on a piece of
content.")]
public bool DeleteTagsOnContent(long contentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagsOnContent", Description = "Deletes all tags on a piece of
content.")>
Public Function DeleteTagsOnContent(contentId As long) As bool
```

Parameters

Parameters	Description
long contentId	The ID of content item.

Returns

True if operation was successful.

Tags.DeleteTagsOnObject Method

Deletes all tags on an object.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagsOnObject", Description = " Deletes all tags on an
object.")]
public bool DeleteTagsOnObject(long objectId, EkEnumeration.CMSObjectTypes objectType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagsOnObject", Description = " Deletes all tags on an
object.")>
Public Function DeleteTagsOnObject(objectId As long, objectType As
EkEnumeration.CMSObjectTypes) As bool
```

Parameters

Parameters	Description
ObjectID	The ID of the object.
ObjectType	The type of object associated with the ObjectId.

Returns

True if operation was successful.

Tags.DeleteTagsOnUser Method

Deletes all tags applied to a user. To delete tags applied BY a given user, use DeleteTagsByUser (see page 1338).

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagsOnUser", Description = "Deletes all tags applied to a
user. To delete tags applied BY a given user, use DeleteTagsByUser")]
public bool DeleteTagsOnUser(long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagsOnUser", Description = "Deletes all tags applied to a
user. To delete tags applied BY a given user, use DeleteTagsByUser")>
Public Function DeleteTagsOnUser(userId As long) As bool
```

Parameters

Parameters	Description
long userId	The ID of the user whose tags are being deleted.

Returns

True if operation was successful

Tags.EditTag Method

Updates an existing Tag.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "EditTag", Description = "Updates an existing Tag.")]
public TagData EditTag(ref TagData tag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "EditTag", Description = "Updates an existing Tag.")>
Public Function EditTag(ByRef tag As TagData) As TagData
```

Parameters

Parameters	Description
ref TagData tag	Tag data to save.

Returns

The ID of a newly created tag or existing tag.

Tags.GetAllTagRelatedContent Method

Returns a list of content related to the supplied Content (see page 1198) via a common Tag. For example, both content item one and content item two are tagged with "CMS" and are considered related.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllTagRelatedContent", Description = "Returns a list of content
related to the supplied Content via a common Tag.  For example, both Content one and Content
Two are both tagged with \"CMS\" and are considered related.")]
public ContentData[] GetAllTagRelatedContent(long contentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllTagRelatedContent", Description = "Returns a list of content
related to the supplied Content via a common Tag.  For example, both Content one and Content
Two are both tagged with \"CMS\" and are considered related.")>
Public Function GetAllTagRelatedContent(contentId As long) As ContentData()
```

Parameters

Parameters	Description
long contentId	This is the content item for which to retrieve related content.

Returns

Returns an array of partially populated ContentData objects.

Tags.GetAllTagRelatedUsers Method

Returns a list of users related to the supplied user via a common Tag. For example, both user one and user two are both tagged with "Engineer" and are considered related.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllTagRelatedUsers", Description = "Returns a list of users
related to the supplied user via a common Tag.  For example, both user one and user Two are
both tagged with \"Engineer\" and are considered related.")]
public UserData[] GetAllTagRelatedUsers(long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllTagRelatedUsers", Description = "Returns a list of users
related to the supplied user via a common Tag.  For example, both user one and user Two are
both tagged with \"Engineer\" and are considered related.")>
Public Function GetAllTagRelatedUsers(userId As long) As UserData()
```

Parameters

Parameters	Description
long userId	This is the user for which to retrieve related users.

Returns

An array of partially populated UserData objects.

GetAllTags Method

This is the overview for the GetAllTags method overload.

Description

This method is overloaded.

- GetAllTags (PagingInfo) - retrieves all tags in the system. This method allows for paging.
- GetAllTags (TagRequestData, Long) - retrieves a list of tags in the system based on the requested information and the total number of tags requested.

Overload List

	Name	Description
	Tags.GetAllTags (PagingInfo) (see page 1343)	Returns a list of all tags in the system. This method can be accessed through the Tags.asmx file located in:
	Tags.GetAllTags (TagRequestData, int) (see page 1343)	Returns a list of all tags in the system. This method can be accessed through the Tags.asmx file located in:

Tags.GetAllTags Method (PagingInfo)

Returns a list of all tags in the system.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllTags", Description = "Returns a list of all tags in the
system.")]
public TagData[] GetAllTags(PagingInfo pagingInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllTags", Description = "Returns a list of all tags in the
system.")>
Public Function GetAllTags(pagingInfo As PagingInfo) As TagData()
```

Parameters

Parameters	Description
PagingInfo pagingInfo	Paging information for call.

Returns

An array of TagData objects.

Tags.GetAllTags Method (TagRequestData, int)

Returns a list of all tags in the system.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllTags with rewuest data", Description = "Returns a list of all
tags in the system.")]
public TagData[] GetAllTags(TagRequestData request, ref int totalTags);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllTags with rewuest data", Description = "Returns a list of all
tags in the system.")>
Public Function GetAllTags(request As TagRequestData, ByRef totalTags As Integer) As TagData()
```

Parameters

Parameters	Description
TagRequestData request	Contains the parameters for the tags to retrieve.
ref int totalTags	Returns the total number of tags available for the requested parameters.

Returns

An array of TagData objects.

GetDefaultTags Method

This is the overview for the GetDefaultTags method overload.

Description

This method is overloaded.

- GetDefaultTags (EkEnumeration.CMSObjectTypes) (see page 1344) - gets default tags for a specified object.
- GetDefaultTags (EkEnumeration.CMSObjectTypes, int) (see page 1345) - gets default tags in a specified language for an object.

Overload List

	Name	Description
	Tags.GetDefaultTags (EkEnumeration.CMSObjectTypes) (see page 1344)	Returns a list of all system defined tags in the system. This method can be accessed through the Tags.asmx file located in:
	Tags.GetDefaultTags (EkEnumeration.CMSObjectTypes, int) (see page 1345)	Returns a list of all system defined tags in the system. This method can be accessed through the Tags.asmx file located in:

Tags.GetDefaultTags Method (EkEnumeration.CMSObjectTypes)

Returns a list of all system defined tags in the system.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDefaultTags", Description = "Returns a list of all system defined
tags in the system.")]
public TagData[] GetDefaultTags(EkEnumeration.CMSObjectTypes objectType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDefaultTags", Description = "Returns a list of all system defined
tags in the system.")>
Public Function GetDefaultTags(objectType As EkEnumeration.CMSObjectTypes) As TagData()
```

Parameters

Parameters	Description
EkEnumeration.CMSObjectTypes objectType	The type of object for which to retrieve tags. For example, retrieve all the default tags for CommunityGroups (CMSObjectTypes.CommunityGroup (see page 1181)).

Returns

Collection (see page 95) of TagData objects

Tags.GetDefaultTags Method (EkEnumeration.CMSObjectTypes, int)

Returns a list of all system defined tags in the system.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDefaultTags with language", Description = "Returns a list of all
system defined tags in the system.")]
public TagData[] GetDefaultTags(EkEnumeration.CMSObjectTypes objectType, int languageId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDefaultTags with language", Description = "Returns a list of all
system defined tags in the system.")>
Public Function GetDefaultTags(objectType As EkEnumeration.CMSObjectTypes, languageId As
Integer) As TagData()
```

Parameters

Parameters	Description
EkEnumeration.CMSObjectTypes objectType	The type of object for which to retrieve tags. For example, retrieve all the default tags for CommunityGroups (CMSObjectTypes.CommunityGroup (see page 1181)).
int languageId	The language of the tags to retrieve.

Returns

Collection (see page 95) of TagData objects.

Tags.GetTagByID Method

Returns the Tag with a specified ID.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagByID", Description = "Returns the Tag with specified Id")]
public TagData GetTagByID(long tagId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagByID", Description = "Returns the Tag with specified Id")>
Public Function GetTagByID(tagId As long) As TagData
```

Parameters

Parameters	Description
long tagId	The ID of the tag to retrieve.

Returns

Returns a tag found with supplied ID or new tag if no tag is found.

Tags.GetTagByName Method

Returns Tag with specified Text.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagByName", Description = "Returns Tag with specified Text")]
public TagData GetTagByName(string tagText);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagByName", Description = "Returns Tag with specified Text")>
Public Function GetTagByName(tagText As string) As TagData
```

Parameters

Parameters	Description
string tagText	Text of the tag to retrieve.

Returns

Returns a tag found with a supplied ID or new tag if no tag is found.

Tags.GetTagCloud Method

Returns a TagCloud (see page 513) based upon the request.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagCloud", Description = "Returns a TagCloud based upon the
request.")]
public TagCloud GetTagCloud(TagCloudRequestData request, ref int totalTags);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagCloud", Description = "Returns a TagCloud based upon the
request.")>
Public Function GetTagCloud(request As TagCloudRequestData, ByRef totalTags As Integer) As
TagCloud
```

Parameters

Parameters	Description
TagCloudRequestData request	Contains the parameters for the tagcloud to retrieve.
ref int totalTags	Returns the total number of tags available in the requested cloud.

Returns

A TagCloud (see page 513) object representing the request.

Tags.GetTagsForContent Method

Returns an array of tags for a piece of content.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagsForContent", Description = "Returns an array of tags for a
piece of content.")]
public TagData[] GetTagsForContent(long contentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagsForContent", Description = "Returns an array of tags for a
piece of content.")>
Public Function GetTagsForContent(contentId As long) As TagData()
```

Parameters

Parameters	Description
long contentId	The ID of the piece of content for which to retrieve tags.

Returns

An array of TagData objects that have been applied to the content.

GetTagsForObject Method

This is the overview for the GetTagsForObject method overload.

Description

This method is overloaded.

- GetTagsForObject (Long, EkEnumeration.CMSObjectsTypes) - returns an array of tags for a given object based on the object's ID and type, and the language ID.
- GetTagsForObject (Long EkEnumeration.CMSObjectTypes, Int) - returns an array of tags for a given object based on the object's ID and type.

Overload List

	Name	Description
	Tags.GetTagsForObject (long, EkEnumeration.CMSObjectTypes) (see page 1347)	Returns an array of tags for a given object. This method can be accessed through the Tags.asmx file located in:
	Tags.GetTagsForObject (long, EkEnumeration.CMSObjectTypes, int) (see page 1348)	Returns an array of tags for a given object. This method can be accessed through the Tags.asmx file located in:

Tags.GetTagsForObject Method (long, EkEnumeration.CMSObjectTypes)

Returns an array of tags for a given object.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagsForObject", Description = "Returns an array of tags for a
given object.")]
public TagData[] GetTagsForObject(long objectId, EkEnumeration.CMSObjectTypes objectType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagsForObject", Description = "Returns an array of tags for a
given object.")>
Public Function GetTagsForObject(objectId As long, objectType As EkEnumeration.CMSObjectTypes)
As TagData()
```

Parameters

Parameters	Description
long objectId	The object ID for which to retrieve tags.
EkEnumeration.CMSObjectTypes objectType	The object type.

Returns

An array of TagData objects that have been applied to an object.

Tags.GetTagsForObject Method (long, EkEnumeration.CMSObjectTypes, int)

Returns an array of tags for a given object.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagsForObject with language)", Description = "Returns an array of
tags for a given object.")]
public TagData[] GetTagsForObject(long objectId, EkEnumeration.CMSObjectTypes objectType, int
languageId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagsForObject with language)", Description = "Returns an array of
tags for a given object.")>
Public Function GetTagsForObject(objectId As long, objectType As EkEnumeration.CMSObjectTypes,
languageId As Integer) As TagData()
```

Parameters

Parameters	Description
long objectId	The object ID for which to retrieve tags.
EkEnumeration.CMSObjectTypes objectType	The object type.
int languageId	The language ID of tags to retrieve.

Returns

An array of TagData objects that have been applied to Object.

GetTagsForUser Method

This is the overview for the GetTagsForUser method overload.

Description

This method is overloaded.

- GetTagsForUser (Long) - returns an array of tags for a user based on the user's ID.
- GetTagsForUser (Long, Int) - returns an array of tags for a specific language associated with a user based on the user's ID and language ID.

Overload List

	Name	Description
	Tags.GetTagsForUser (long) (see page 1349)	Returns an array of tags for a user. This method can be accessed through the Tags.asmx file located in:
	Tags.GetTagsForUser (long, int) (see page 1349)	Returns an array of tags for a user. This method can be accessed through the Tags.asmx file located in:

Tags.GetTagsForUser Method (long)

Returns an array of tags for a user.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagsForUser", Description = "Returns an array of tags for a
user.")]
public TagData[] GetTagsForUser(long userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagsForUser", Description = "Returns an array of tags for a
user.")>
Public Function GetTagsForUser(userId As long) As TagData()
```

Parameters

Parameters	Description
long userId	The ID of the user for which to retrieve tags.

Returns

An array of TagData objects that have been applied to the user.

Tags.GetTagsForUser Method (long, int)

Returns an array of tags for a user.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "GetTagsForUser with language)", Description = "Returns an array of
tags for a user.")]
public TagData[] GetTagsForUser(long userId, int languageId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagsForUser with language)", Description = "Returns an array of
tags for a user.")>
Public Function GetTagsForUser(userId As long, languageId As Integer) As TagData()
```

Parameters

Parameters	Description
long userId	The ID of the user for which to retrieve tags.
int languageId	Language of tags to retrieve.

Returns

An array of TagData objects that have been applied to the user.

Tags.RemoveTagAsDefault Method

Removes a tag from the object's default tag list.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RemoveTagAsDefault", Description = "")]
public void RemoveTagAsDefault(string tagText, int languageId, EkEnumeration.CMSObjectTypes
objectType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RemoveTagAsDefault", Description = "")>
Public Function RemoveTagAsDefault(tagText As string, languageId As Integer, objectType As
EkEnumeration.CMSObjectTypes) As void
```

Parameters

Parameters	Description
string tagText	The text of the tag.
int languageId	The ID of the tag being removed.
EkEnumeration.CMSObjectTypes objectType	The type of object.

Tags.SaveTagAsDefault Method

Saves a tag to the the object's default list.

This method can be accessed through the Tags.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Community/Tags.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SaveTagAsDefault", Description = "")]
public TagData SaveTagAsDefault(string tagText, int languageId, EkEnumeration.CMSObjectTypes
objectType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SaveTagAsDefault", Description = "")>
Public Function SaveTagAsDefault(tagText As string, languageId As Integer, objectType As
EkEnumeration.CMSObjectTypes) As TagData
```

Parameters

Parameters	Description
string tagText	The text of the tag.
int languageId	The ID of the language to which the tag will be assigned.
EkEnumeration.CMSObjectTypes objectType	The type of object.

Returns

TagData structure representing the tag saved as default.

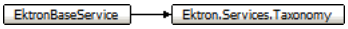
Taxonomy Class

This section describes the Web Service methods used with Ektron's Taxonomy. The Taxonomy Web service is a wrapper for the Taxonomy API (see page 669): `Ektron.Cms.API.Content.Taxonomy` (see page 876). These methods can be accessed through the Taxonomy.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Taxonomy : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Taxonomy
Inherits EktronBaseService
```

File

Taxonomy.cs

Taxonomy Methods

The methods of the Taxonomy class are listed here.

Public Methods

	Name	Description
	AddTaxonomyItem (see page 1352)	This is the overview for the AddTaxonomyItem method overload.
	AddTaxonomySyncFolder (see page 1353)	Adds a folder entry into the taxonomy. This method can be accessed through the Taxonomy.asmx file located in:
	CreateTaxonomy (see page 1353)	Creates a new taxonomy. This method can be accessed through the Taxonomy.asmx file located in:

	DeleteTaxonomy (see page 1354)	Deletes the existing taxonomy. This method can be accessed through the Taxonomy.asmx file located in:
	GetTaxonomyIdByPath (see page 1354)	Returns the taxonomy ID by a given path. This method can be accessed through the Taxonomy.asmx file located in:
	LoadTaxonomy (see page 1355)	Returns the TaxonomyData with all its subcategories and its items. This method can be accessed through the Taxonomy.asmx file located in:
	ReadAllAssignedCategory (see page 1355)	Returns all the assigned category for the content ID. This method can be accessed through the Taxonomy.asmx file located in:
	ReadTaxonomy (see page 1356)	Read the taxonomy data. This method can be accessed through the Taxonomy.asmx file located in:
	RemoveTaxonomyFolder (see page 1356)	Removes the sync up folder from the taxonomy. This method can be accessed through the Taxonomy.asmx file located in:
	RemoveTaxonomyItem (see page 1357)	Removes an item from the taxonomy. This method can be accessed through the Taxonomy.asmx file located in:
	ReOrderTaxonomyItems (see page 1357)	Reorder the display of taxonomy items. This method can be accessed through the Taxonomy.asmx file located in:
	UpdateTaxonomy (see page 1357)	Updates the existing taxonomy. This method can be accessed through the Taxonomy.asmx file located in:

AddTaxonomyItem Method

This is the overview for the AddTaxonomyItem method overload.

Description

This method is overloaded.

- AddTaxonomyItem (TaxonomyContentRequest) (see page 1352)
- AddTaxonomyItem (TaxonomyRequest) (see page 1353)

Overload List

	Name	Description
	Taxonomy.AddTaxonomyItem (TaxonomyContentRequest) (see page 1352)	Inserts an item into the taxonomy. This method can be accessed through the Taxonomy.asmx file located in:
	Taxonomy.AddTaxonomyItem (TaxonomyRequest) (see page 1353)	Inserts an item into the taxonomy. This method can be accessed through the Taxonomy.asmx file located in:

Taxonomy.AddTaxonomyItem Method (TaxonomyContentRequest)

Inserts an item into the taxonomy.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTaxonomyItem ContentRequest", Description = "inserts item into
taxonomy")]
public void AddTaxonomyItem(TaxonomyContentRequest TaxonomyReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTaxonomyItem ContentRequest", Description = "inserts item into
taxonomy")>
```

```
Public Function AddTaxonomyItem(TaxonomyReq As TaxonomyContentRequest) As void
```

Parameters

Parameters	Description
TaxonomyContentRequest TaxonomyReq	taxonomycontentrequest

Taxonomy.AddTaxonomyItem Method (TaxonomyRequest)

Inserts an item into the taxonomy.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTaxonomyItem", Description = "inserts item into taxonomy")]
public void AddTaxonomyItem(TaxonomyRequest TaxonomyReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTaxonomyItem", Description = "inserts item into taxonomy")>
Public Function AddTaxonomyItem(TaxonomyReq As TaxonomyRequest) As void
```

Parameters

Parameters	Description
TaxonomyRequest TaxonomyReq	taxonomyrequest

Taxonomy.AddTaxonomySyncFolder Method

Adds a folder entry into the taxonomy.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTaxonomySyncFolder", Description = "adds folder entry")]
public void AddTaxonomySyncFolder(TaxonomySyncRequest TaxonomyReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTaxonomySyncFolder", Description = "adds folder entry")>
Public Function AddTaxonomySyncFolder(TaxonomyReq As TaxonomySyncRequest) As void
```

Parameters

Parameters	Description
TaxonomySyncRequest TaxonomyReq	taxonomyrequest

Taxonomy.CreateTaxonomy Method

Creates a new taxonomy.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CreateTaxonomy", Description = "creates new taxonomy")]
public long CreateTaxonomy(TaxonomyData TaxonomyReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CreateTaxonomy", Description = "creates new taxonomy")>
Public Function CreateTaxonomy(TaxonomyReq As TaxonomyData) As long
```

Parameters

Parameters	Description
TaxonomyData TaxonomyReq	taxonomy data

Returns

new taxonomy ID

Taxonomy.DeleteTaxonomy Method

Deletes the existing taxonomy.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTaxonomy", Description = "deletes the existing taxonomy")]
public void DeleteTaxonomy(TaxonomyRequest TaxonomyReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTaxonomy", Description = "deletes the existing taxonomy")>
Public Function DeleteTaxonomy(TaxonomyReq As TaxonomyRequest) As void
```

Parameters

Parameters	Description
TaxonomyRequest TaxonomyReq	taxonomy request

Taxonomy.GetTaxonomyIdByPath Method

Returns the taxonomy ID by a given path.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaxonomyIdByPath", Description = "returns taxonomy id by given
path")]
public long GetTaxonomyIdByPath(string TaxonomyPath);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "GetTaxonomyIdByPath", Description = "returns taxonomy id by given
path")>
Public Function GetTaxonomyIdByPath(TaxonomyPath As string) As long
```

Parameters

Parameters	Description
path	the path to the taxonomy

Returns

taxonomy id

Taxonomy.LoadTaxonomy Method

Returns the TaxonomyData with all its subcategories and its items.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "LoadTaxonomy", Description = "returns taxonomy id by given path")]
public TaxonomyData LoadTaxonomy(ref TaxonomyRequest TaxonomyReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "LoadTaxonomy", Description = "returns taxonomy id by given path")>
Public Function LoadTaxonomy(ByRef TaxonomyReq As TaxonomyRequest) As TaxonomyData
```

Parameters

Parameters	Description
ref TaxonomyRequest TaxonomyReq	TaxonomyRequest (see page 1643)

Returns

TaxonomyData

Taxonomy.ReadAllAssignedCategory Method

Returns all the assigned category for the content ID.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ReadAllAssignedCategory", Description = "returns all the assigned
category for the content id")]
public TaxonomyBaseData[] ReadAllAssignedCategory(long ContentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ReadAllAssignedCategory", Description = "returns all the assigned
category for the content id")>
Public Function ReadAllAssignedCategory(ContentId As long) As TaxonomyBaseData()
```

Parameters

Parameters	Description
contentid	the ID for which the categories are returned

Returns

An array of taxonomybasedata

Taxonomy.ReadTaxonomy Method

Read the taxonomy data.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ReadTaxonomy", Description = "read the taxonomy data")]
public TaxonomyData ReadTaxonomy(ref TaxonomyRequest TaxonomyReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ReadTaxonomy", Description = "read the taxonomy data")>
Public Function ReadTaxonomy(ByRef TaxonomyReq As TaxonomyRequest) As TaxonomyData
```

Parameters

Parameters	Description
ref TaxonomyRequest TaxonomyReq	taxonomyrequest

Returns

taxonomydata

Taxonomy.RemoveTaxonomyFolder Method

Removes the sync up folder from the taxonomy.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RemoveTaxonomyFolder", Description = "removes the sync up folder
from taxonomy")]
public void RemoveTaxonomyFolder(TaxonomySyncRequest TaxonomyReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RemoveTaxonomyFolder", Description = "removes the sync up folder
from taxonomy")>
Public Function RemoveTaxonomyFolder(TaxonomyReq As TaxonomySyncRequest) As void
```

Parameters

Parameters	Description
TaxonomySyncRequest TaxonomyReq	taxonomyrequest

Taxonomy.RemoveTaxonomyItem Method

Removes an item from the taxonomy.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RemoveTaxonomyItem", Description = "removes item from the taxonomy")]
public void RemoveTaxonomyItem(TaxonomyRequest TaxonomyReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RemoveTaxonomyItem", Description = "removes item from the taxonomy")>
Public Function RemoveTaxonomyItem(TaxonomyReq As TaxonomyRequest) As void
```

Parameters

Parameters	Description
TaxonomyRequest TaxonomyReq	taxonomyrequest

Taxonomy.ReOrderTaxonomyItems Method

Reorder the display of taxonomy items.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ReOrderTaxonomyItems", Description = "reorder display of taxonomy
items")]
public void ReOrderTaxonomyItems(TaxonomyRequest TaxonomyReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ReOrderTaxonomyItems", Description = "reorder display of taxonomy
items")>
Public Function ReOrderTaxonomyItems(TaxonomyReq As TaxonomyRequest) As void
```

Parameters

Parameters	Description
TaxonomyRequest TaxonomyReq	taxonomyrequest

Taxonomy.UpdateTaxonomy Method

Updates the existing taxonomy.

This method can be accessed through the Taxonomy.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Taxonomy/Taxonomy.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateTaxonomy", Description = "updates the existing taxonomy")]
```

```
public void UpdateTaxonomy(TaxonomyData TaxonomyReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateTaxonomy", Description = "updates the existing taxonomy")>
Public Function UpdateTaxonomy(TaxonomyReq As TaxonomyData) As void
```

Parameters

Parameters	Description
TaxonomyData TaxonomyReq	taxonomydata

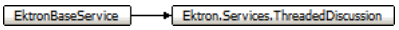
ThreadedDiscussion Class

This section describes the Web Service methods used with Ektron's Threaded Discussion feature. The Threaded Discussion Web service is a wrapper for the Threaded Discussion API (see page 669): `Ektron.Cms.API.Content.ThreadedDiscussion` (see page 882). These methods can be accessed through the ThreadedDiscussion.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header.

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class ThreadedDiscussion : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class ThreadedDiscussion
Inherits EktronBaseService
```

File

ThreadedDiscussion.cs

ThreadedDiscussion Methods

The methods of the ThreadedDiscussion class are listed here.

Public Methods

	Name	Description
	AddBoard (see page 1359)	Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns numeric ID of the new discussion board. This method can be accessed through the ThreadedDiscussion.asmx file located in:
	AddForum (see page 1360)	Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns the numeric ID of new discussion forum. This method can be accessed through the ThreadedDiscussion.asmx file located in:
	AddReply (see page 1360)	Adds a Reply to the Forum (see page 264).
	AddTopic (see page 1361)	This is the overview for the AddTopic method overload.
	DeleteBoard (see page 1362)	Deletes a board with a given ID. This will delete the board as well as all forums, topics and posts under it. BoardID must be greater than 0. This method can be accessed through the ThreadedDiscussion.asmx file located in:

	DeleteForum (see page 1363)	Deletes a forum with a given ID. This will delete the forum as well as all topics and posts under it. forumID must be greater than 0. This method can be accessed through the ThreadedDiscussion.asmx file located in:
	DeleteTopic (see page 1363)	Deletes a topic by topicID. This method deletes all posts associated with the topic as well. Returns boolean value. This method can be accessed through the ThreadedDiscussion.asmx file located in:
	GetBoardCategories (see page 1364)	Method to get the Categories for a board by ID. Returns either zero length or populated array. See DiscussionCategory (see page 1614) class for more information. This method can be accessed through the ThreadedDiscussion.asmx file located in:
	GetBoardForums (see page 1364)	Method to get the Forums for a board by ID. Returns either zero length or populated array. See DiscussionForum class for more information. This method can be accessed through the ThreadedDiscussion.asmx file located in:
	GetDiscussionBoard (see page 1364)	This is the overview for the GetDiscussionBoard method overload.
	GetDiscussionBoardString (see page 1366)	Method to get the rendered view of a discussion board by ID. Returns a string of the output based upon the input parameters. Display of the board can be controlled by using CSS stylesheets. This method can be accessed through the ThreadedDiscussion.asmx file located in:
	GetForum (see page 1366)	Method to obtain a particular forum by ID. Will return a discussion board with a 1 length Discussion Forum (see page 264) array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned. This method can be accessed through the ThreadedDiscussion.asmx file located in:
	GetRepliesForTopics (see page 1367)	Method returns a list of replies for a given forum ID. **ArrayList Structure** • ArrayListItem(j)(0) ' content_id • ArrayListItem(j)(1) ' comment_total • ArrayListItem(j)(2) ' comment_sum • ArrayListItem(j)(3) ' last_posted • ArrayListItem(j)(4) ' content_language This method can be accessed through the ThreadedDiscussion.asmx file located in:
	GetTopic (see page 1368)	Method to obtain a particular Topic by ID. Will return a discussion board with a 1 length Discussion Forum (see page 264) array containing a 1 length DiscussionTopic array. If the topic does not exist, then a DiscussionBoard with a 1 length Forums property containing a zero length Topics property will be returned. This method can be accessed through the ThreadedDiscussion.asmx file located in:
	SearchReplies (see page 1368)	Method to obtain an array of topic replies matching search criteria. Will return an instance of EkTasks, which is effectively an array of EkTask objects. If no results are returned, then a zero length EkTasks will be returned. This method can be accessed through the ThreadedDiscussion.asmx file located in:

ThreadedDiscussion.AddBoard Method

Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns numeric ID of the new discussion board.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddBoard", Description = "")]
public long AddBoard(long ParentID, string Name, string Title, bool RequiresAuthentication,
string StyleSheet, string[] Categories);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddBoard", Description = "")>
Public Function AddBoard(ParentID As long, Name As string, Title As string,
RequiresAuthentication As bool, StyleSheet As string, Categories As string[]) As long
```

Parameters

Parameters	Description
long ParentID	The numeric ID of the folder to which the discussion board is being added.
string Name	Name of the discussion board.
string Title	Title of the discussion board.
bool RequiresAuthentication	Enter True if the discussion board requires authentication. If it does not require authentication, enter false.
string StyleSheet	The stylesheet to apply to the discussion board.
string[] Categories	String array of categories for the discussion board.

Returns

Returns a numeric ID of new discussion board (Long).

ThreadedDiscussion.AddForum Method

Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns the numeric ID of new discussion forum.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddForum", Description = "")]
public long AddForum(long BoardID, string Name, string Description, bool ModeratePosts, bool
LockForum, int SortOrder, long CategoryID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddForum", Description = "")>
Public Function AddForum(BoardID As long, Name As string, Description As string, ModeratePosts
As bool, LockForum As bool, SortOrder As Integer, CategoryID As long) As long
```

Parameters

Parameters	Description
long BoardID	The numeric ID of the discussion board to which you want to add a forum.
string Name	The name of forum.
string Description	String value for the forum summary.
bool ModeratePosts	Are posts moderated?
bool LockForum	Is the forum locked?
int SortOrder	Sort order for the new forum.
long CategoryID	The numeric value of the Category ID the new forum belongs to.

Returns

Returns the numeric ID of the new forum (Long).

ThreadedDiscussion.AddReply Method

Adds a Reply to the Forum (see page 264).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddReply", Description = "Adds a reply")]
public bool AddReply(ref TaskData data);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddReply", Description = "Adds a reply")>
Public Function AddReply(ByRef data As TaskData) As bool
```

Parameters

Parameters	Description
ref TaskData data	TaskData

Returns

False=if successfully added otherwise True

AddTopic Method

This is the overview for the AddTopic method overload.

Description

This method is overloaded.

- AddTopic (Long, String, String) - add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as Long. This method is overloaded.
- AddTopic (Long, String, String, Int) - add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as Long.

Overload List

	Name	Description
	ThreadedDiscussion.AddTopic (long, string, string) (see page 1361)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns a numeric topic ID (Long). This method can be accessed through the ThreadedDiscussion.asmx file located in:
	ThreadedDiscussion.AddTopic (long, string, string, int) (see page 1362)	Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns a numeric Topic ID (Long). This method can be accessed through the ThreadedDiscussion.asmx file located in:

ThreadedDiscussion.AddTopic Method (long, string, string)

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns a numeric topic ID (Long).

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTopic", Description = "")]
public long AddTopic(long ForumID, string Title, string Message);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTopic", Description = "")>
Public Function AddTopic(ForumID As long, Title As string, Message As string) As long
```

Parameters

Parameters	Description
long ForumID	A numeric ID of the forum to add topic to.
string Title	String value of the topic title.
string Message	String value of the topic message.

Returns

A numeric value for the topic ID (Long).

ThreadedDiscussion.AddTopic Method (long, string, string, int)

Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns a numeric Topic ID (Long).

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTopic With Language ID", Description = "")]
public long AddTopic(long ForumID, string Title, string Message, int LanguageId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTopic With Language ID", Description = "")>
Public Function AddTopic(ForumID As long, Title As string, Message As string, LanguageId As
Integer) As long
```

Parameters

Parameters	Description
long ForumID	The numeric ID of the forum to add topic to.
string Title	String value of the topic title.
string Message	String value of the topic message.
int LanguageId	The numeric ID of the Language ID.

Returns

Returns a numeric value for the topic ID (Long).

ThreadedDiscussion.DeleteBoard Method

Deletes a board with a given ID. This will delete the board as well as all forums, topics and posts under it. BoardID must be greater than 0.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteBoard", Description = "")]
public void DeleteBoard(long BoardID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "DeleteBoard", Description = "")>
Public Function DeleteBoard(BoardID As long) As void
```

Parameters

Parameters	Description
long BoardID	The numeric ID of the board. This corresponds to the folder ID.

ThreadedDiscussion.DeleteForum Method

Deletes a forum with a given ID. This will delete the forum as well as all topics and posts under it. forumID must be greater than 0.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteForum", Description = "")]
public void DeleteForum(long forumID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteForum", Description = "")>
Public Function DeleteForum(forumID As long) As void
```

Parameters

Parameters	Description
long forumID	The numeric ID of the forum. This corresponds to the folder ID.

ThreadedDiscussion.DeleteTopic Method

Deletes a topic by topicID. This method deletes all posts associated with the topic as well. Returns boolean value.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTopic", Description = "")]
public bool DeleteTopic(long topicid);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTopic", Description = "")>
Public Function DeleteTopic(topicid As long) As bool
```

Parameters

Parameters	Description
post_id	Numeric value of the topic you wish to delete.

Returns

Boolean

ThreadedDiscussion.GetBoardCategories Method

Method to get the Categories for a board by ID. Returns either zero length or populated array. See DiscussionCategory (see page 1614) class for more information.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBoardCategories", Description = "")]
public DiscussionCategory[] GetBoardCategories(long boardID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBoardCategories", Description = "")>
Public Function GetBoardCategories(boardID As long) As DiscussionCategory()
```

Parameters

Parameters	Description
long boardID	The numeric value of the board ID.

Returns

Returns an array of type DiscussionCategory (see page 1614).

ThreadedDiscussion.GetBoardForums Method

Method to get the Forums for a board by ID. Returns either zero length or populated array. See DiscussionForum class for more information.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBoardForums", Description = "")]
public DiscussionForum[] GetBoardForums(long boardID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBoardForums", Description = "")>
Public Function GetBoardForums(boardID As long) As DiscussionForum()
```

Parameters

Parameters	Description
long boardID	The numeric value of the board ID.

Returns

Returns an array of type DiscussionForum.

GetDiscussionBoard Method

This is the overview for the GetDiscussionBoard method overload.

Description

This method is overloaded.

- GetDiscussionBoard (Long) - get a discussion board by ID or path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.
- GetDiscussionBoard (String) - get a discussion board by path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

Overload List

	Name	Description
	ThreadedDiscussion.GetDiscussionBoard (long) (see page 1365)	Method to get a board by ID. Returns a DiscussionBoard class for a given ID value. See DiscussionBoard class for more information on the properties of the returned object. This method can be accessed through the ThreadedDiscussion.asmx file located in:
	ThreadedDiscussion.GetDiscussionBoard (string) (see page 1365)	Method to get a discussion board by path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object. This method can be accessed through the ThreadedDiscussion.asmx file located in:

ThreadedDiscussion.GetDiscussionBoard Method (long)

Method to get a board by ID. Returns a DiscussionBoard class for a given ID value. See DiscussionBoard class for more information on the properties of the returned object.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDiscussionBoard By Board ID", Description = "")]
public DiscussionBoard GetDiscussionBoard(long boardID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDiscussionBoard By Board ID", Description = "")>
Public Function GetDiscussionBoard(boardID As long) As DiscussionBoard
```

Parameters

Parameters	Description
long boardID	The numeric value of the board ID.

Returns

Returns a class of type DiscussionBoard.

ThreadedDiscussion.GetDiscussionBoard Method (string)

Method to get a discussion board by path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDiscussionBoard By Board Path", Description = "")]
public DiscussionBoard GetDiscussionBoard(string boardpath);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDiscussionBoard By Board Path", Description = "")>
Public Function GetDiscussionBoard(boardpath As string) As DiscussionBoard
```

Parameters

Parameters	Description
string boardpath	String value of the board path. Ex: \Ektron\CMS400.NET.

Returns

Returns a class of type DiscussionBoard.

ThreadedDiscussion.GetDiscussionBoardString Method

Method to get the rendered view of a discussion board by ID. Returns a string of the output based upon the input parameters. Display of the board can be controlled by using CSS stylesheets.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "", Description = "")]
public string GetDiscussionBoardString(long boardID, string URLpath);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "", Description = "")>
Public Function GetDiscussionBoardString(boardID As long, URLpath As string) As string
```

Parameters

Parameters	Description
long boardID	The numeric value of the board ID.
string URLpath	String value of the URL path used. This is generally the Server Variable "URL". Ex: "/forum.aspx".

Returns

Returns a string of the rendered Discussion Board, suitable for the server control or a Web service.

ThreadedDiscussion.GetForum Method

Method to obtain a particular forum by ID. Will return a discussion board with a 1 length Discussion Forum (see page 264) array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetForum", Description = "")]
public DiscussionBoard GetForum(long forumid);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetForum", Description = "")>
Public Function GetForum(forumid As long) As DiscussionBoard
```

Parameters

Parameters	Description
long forumid	The numeric value of the forum ID.

Returns

An instance of type DiscussionBoard.

ThreadedDiscussion.GetRepliesForTopics Method

Method returns a list of replies for a given forum ID.

ArrayList Structure

- ArrayListItem(j)(0) ' content_id
- ArrayListItem(j)(1) ' comment_total
- ArrayListItem(j)(2) ' comment_sum
- ArrayListItem(j)(3) ' last_posted
- ArrayListItem(j)(4) ' content_language

This method can be accessed through the ThreadedDiscussion.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetRepliesForTopics", Description = "")]
public ArrayList GetRepliesForTopics(long ForumID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetRepliesForTopics", Description = "")>
Public Function GetRepliesForTopics(ForumID As long) As ArrayList
```

Parameters

Parameters	Description
long ForumID	The numeric value of the forum ID.

Returns

An instance of type ArrayList.

Remarks

This is an older method and is deprecated. The information returned herein was merged into the higher performance GetForum (see page 1366)(int forumID) method, and that should be used instead.

ThreadedDiscussion.GetTopic Method

Method to obtain a particular Topic by ID. Will return a discussion board with a 1 length Discussion Forum (see page 264) array containing a 1 length DiscussionTopic array. If the topic does not exist, then a DiscussionBoard with a 1 length Forums property containing a zero length Topics property will be returned.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTopic", Description = "")]
public DiscussionBoard GetTopic(long topicid);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTopic", Description = "")>
Public Function GetTopic(topicid As long) As DiscussionBoard
```

Parameters

Parameters	Description
long topicid	The numeric value of the Topic ID.

Returns

An instance of type DiscussionBoard.

ThreadedDiscussion.SearchReplies Method

Method to obtain an array of topic replies matching search criteria. Will return an instance of EkTasks, which is effectively an array of EkTask objects. If no results are returned, then a zero length EkTasks will be returned.

This method can be accessed through the ThreadedDiscussion.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/Content/ThreadedDiscussion.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SearchReplies", Description = "")]
public EkTasks SearchReplies(long[] ForumIDs, string[] SearchTerms, EkEnumeration.SearchTypes
stType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SearchReplies", Description = "")>
Public Function SearchReplies(ForumIDs As long[], SearchTerms As string[], stType As
EkEnumeration.SearchTypes) As EkTasks
```

Parameters

Parameters	Description
long[] ForumIDs	A numeric (Long) array of the Forum (see page 264) IDs you want to search. Use zero length array, or 0, to search all.
string[] SearchTerms	String array of the search terms (case insensitive).
EkEnumeration.SearchTypes stType	EkEnumeration.SearchTypes enumeration value for the search type.

Returns

An instance of type ekTasks (an array of Tasks).

User Class

This section describes the Web Service methods used with Users (see page 1076). The Users (see page 1076) Web service is a wrapper for the User API (see page 669): `Ektron.Cms.API.User.User` (see page 958). These methods can be accessed through the User.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/User/User.asmx`

Web Services (see page 1083) examples involving content are available in the Developer sample site. You can download this site from:

http://www.ektron.com/cms400-web-cms.aspx?id=5278

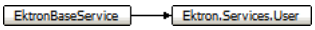
Once the developer site is installed, example code files are located in your site root. For example:

`C:\Inetpub\wwwroot\CMS400Developer\Developer\dotNetAssemblies\WebServicesAPI\UserAPI`

IMPORTANT: If you are accessing these Web Services (see page 1083) remotely, you will need to set the authentication header. For example:

```
userAPI.AuthenticationHeaderValue = New localhost.AuthenticationHeader()
userAPI.AuthenticationHeaderValue.Username = "admin"
userAPI.AuthenticationHeaderValue.Password = "admin"
userAPI.AuthenticationHeaderValue.Domain = "Domain" '''Only needed if AD/LDAP is being used
```

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class User : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class User
Inherits EktronBaseService
```

File

User.cs

User Methods

The methods of the User class are listed here.

Public Methods

	Name	Description
	ActivateUserAccount (see page 1371)	This is the overview for the ActivateUserAccount method overload.
	AddMembershipUser (see page 1373)	Creates new membership user. This method can be accessed through the User.asmx file located in:
	AddNewUser (see page 1373)	Adds a new user to the CMS. This method can be accessed through the User.asmx file located in:

	AddRolePermission (see page 1374)	Adds permission for the given role. This method can be accessed through the User.asmx file located in:
	AddUser (see page 1374)	Creates new cms user. This method can be accessed through the User.asmx file located in:
	AddUserGroup (see page 1375)	Creates new user group. This method can be accessed through the User.asmx file located in:
	AddUserToGroup (see page 1375)	Adds an existing user to an existing group. This method can be accessed through the User.asmx file located in:
	AutoLogInUser (see page 1376)	Login (see page 347) the given user credentials. This method can be accessed through the User.asmx file located in:
	CreateBlankUserObject (see page 1376)	This is the overview for the CreateBlankUserObject method overload.
	DeleteUserByID (see page 1377)	Removes the user from the system. This will not purge the data, but it disables the user from further accessing the system. This method can be accessed through the User.asmx file located in:
	DeleteUserByIds (see page 1378)	Removes the user from the system. This will not purge the data, but it disables the user from further accessing the system. This method can be accessed through the User.asmx file located in:
	DeleteUserFromGroup (see page 1378)	Removes a user from the designated group. This method can be accessed through the User.asmx file located in:
	DeleteUserGroup (see page 1379)	Removes the group from the system permanently. This method can be accessed through the User.asmx file located in:
	EditUserCustomProperties (see page 1379)	This method creates the editable fields(UI) for user's custom properties. This method can be accessed through the User.asmx file located in:
	GetActiveUser (see page 1380)	Returns all of the user properties with or without preference. This method can be accessed through the User.asmx file located in:
	GetADMapping (see page 1380)	Loads the CMS and Active Directory (see page 196) name. This method can be accessed through the User.asmx file located in:
	GetAllCmsUsers (see page 1381)	Loads all the CMS users. This method can be accessed through the User.asmx file located in:
	GetAllCustomProperties (see page 1381)	Returns array of all the custom properties defined in the system. This method can be accessed through the User.asmx file located in:
	GetAllUserByCustomProperty (see page 1382)	Returns all the users based on the given properties. This method can be accessed through the User.asmx file located in:
	GetAllUserGroups (see page 1382)	This is the overview for the GetAllUserGroups method overload.
	GetAllUsers (see page 1384)	This is the overview for the GetAllUsers method overload.
	GetGroupsUserIsIn (see page 1385)	Loads all the groups by user. This method can be accessed through the User.asmx file located in:
	GetMembershipUser (see page 1386)	Returns user data based on the user's login name (username). This method can be accessed through the User.asmx file located in:
	GetUser (see page 1386)	Loads the user details for the given user. This method can be accessed through the User.asmx file located in:
	GetUserbyUsername (see page 1387)	Loads the user details for the given user. This method can be accessed through the User.asmx file located in:
	GetUserCustomProperties (see page 1387)	Returns an array of custom properties for a given user's ID. This method can be accessed through the User.asmx file located in:
	GetUserEmailInfo (see page 1388)	Returns user data for a given user's ID. This method can be accessed through the User.asmx file located in:
	GetUserForFolderAdmin (see page 1388)	
	GetUserGroup (see page 1389)	Loads a given group's details. This method can be accessed through the User.asmx file located in:
	GetUserGroupForFolderAdmin (see page 1389)	Returns all the groups data for the administrator. This method can be accessed through the User.asmx file located in:
	GetUserPermissions (see page 1390)	Loads the user permissions as a array of UserPermissionData. This method can be accessed through the User.asmx file located in:
	GetUserPreference (see page 1390)	Loads the preferences for the given user. This method can be accessed through the User.asmx file located in:
	GetUsers (see page 1391)	This is the overview for the GetUsers method overload.

	GetUsersForTask (see page 1393)	Loads the users that have permissions to the given content. This method can be accessed through the User.asmx file located in:
	GetUsersNotInGroup (see page 1393)	Loads all the users that are not in the given group. This method can be accessed through the User.asmx file located in:
	IsAdmin (see page 1394)	Determines the if the current user is an Administrator. This method can be accessed through the User.asmx file located in:
	IsARoleMember (see page 1394)	This is the overview for the IsARoleMember method overload.
	IsARoleMemberForContent (see page 1398)	Determines whether a Role Member is allowed to work with a given piece of content as an administrator or not. Administrations privileges are for the given piece of content only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This method can be accessed through the User.asmx file located in:
	IsARoleMemberForFolder (see page 1398)	Determines whether a Role Member is allowed to work with a given folder as an administrator or not. Administrations privileges are for the given folder only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This method can be accessed through the User.asmx file located in:
	LogInUser (see page 1399)	Used to log in the user. This method can be accessed through the User.asmx file located in:
	ResetMembershipUserPassword (see page 1400)	This is the overview for the ResetMembershipUserPassword method overload.
	UnsubscribeUser (see page 1401)	Unsubscribes a user from the system. This method can be accessed through the User.asmx file located in:
	UpdateCustomPropertiesItemOrder (see page 1401)	Updates the order of available custom properties. This method can be accessed through the User.asmx file located in:
	UpdateMapCoordinate (see page 1402)	Sets the map co-ordinate for the given user property. This method can be accessed through the User.asmx file located in:
	UpdateUser (see page 1402)	This is the overview for the UpdateUser method overload.
	UpDateUserGroup (see page 1404)	Updates the user group details. This method can be accessed through the User.asmx file located in:
	UpdateUserPreferences (see page 1404)	Updates the given user's preference data. This method can be accessed through the User.asmx file located in:
	UserObject (see page 1405)	This is the overview for the UserObject method overload.

ActivateUserAccount Method

This is the overview for the ActivateUserAccount method overload.

Description

This method is overloaded.

- ActivateUserAccount (int) - activates a deleted (deactivated) user into the current/active state using the UserId.
- ActivateUserAccount (string, string) (see page 1372) - activates the deleted (deactivated) user into the current/active state using the Username and GUID (AccountID) parameter.

Overload List

	Name	Description
	User.ActivateUserAccount (long) (see page 1372)	Activates a deleted (deactivated) user into the current/active state using the UserId. This function is overloaded. See Also: ActivateUserAccount(String, String) This method can be accessed through the User.asmx file located in:
	User.ActivateUserAccount (string, string) (see page 1372)	Activates the deleted (deactivated) user into the current/active state using the Username and GUID (AccountID) parameter. This function is overloaded. See Also: ActivateUserAccount(Long) This method can be accessed through the User.asmx file located in:

User.ActivateUserAccount Method (long)

Activates a deleted (deactivated) user into the current/active state using the UserId.

This function is overloaded. See Also: ActivateUserAccount(String, String)

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ActivateUserAccount By UserID", Description = "")]
public bool ActivateUserAccount(long UserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ActivateUserAccount By UserID", Description = "")>
Public Function ActivateUserAccount(UserId As long) As bool
```

Parameters

Parameters	Description
long UserId	A unique ID for the required user

Returns

Boolean

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.ActivateUserAccount Method (string, string)

Activates the deleted (deactivated) user into the current/active state using the Username and GUID (AccountID) parameter.

This function is overloaded. See Also: ActivateUserAccount(Long)

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ActivateUserAccount By UserName and AccountID", Description = "")]
public bool ActivateUserAccount(string Username, string AccountID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ActivateUserAccount By UserName and AccountID", Description = "")>
Public Function ActivateUserAccount(Username As string, AccountID As string) As bool
```

Parameters

Parameters	Description
string Username	The login name for the user
string AccountID	The GUID for the user

Returns

Boolean

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.AddMembershipUser Method

Creates new membership user.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddMembershipUser", Description = "")]
public UserData AddMembershipUser(UserData UserData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddMembershipUser", Description = "")>
Public Function AddMembershipUser(UserData As UserData) As UserData
```

Parameters

Parameters	Description
UserData UserData	An object of class which represents the corresponding user elements

Returns

Ektron.Cms.UserData

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.AddNewUser Method

Adds a new user to the CMS.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddNewUser", Description = "")]
[Obsolete("Use \"AddMembershipUser\" instead.")]
public UserData AddNewUser(UserData UserData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddNewUser", Description = "")>
<Obsolete("Use \"AddMembershipUser\" instead.")>
Public Function AddNewUser(UserData As UserData) As UserData
```

Parameters

Parameters	Description
UserData UserData	An object of class which represents the corresponding user elements

Returns

Ektron.Cms.UserData

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.AddRolePermission Method

Adds permission for the given role.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddRolePermission", Description = "")]
public bool AddRolePermission(string RoleName);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddRolePermission", Description = "")>
Public Function AddRolePermission(RoleName As string) As bool
```

Parameters

Parameters	Description
string RoleName	Required role name.

Returns

True if successfully added. Otherwise, returns false.

User.AddUser Method

Creates new cms user.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddUser", Description = "Creates new cms user.")]
public UserData AddUser(UserData user);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddUser", Description = "Creates new cms user.")>
Public Function AddUser(user As UserData) As UserData
```

Parameters

Parameters	Description
UserData user	Collection (see page 95) of User (see page 1369)'s info.

Returns

Ektron.Cms.UserData

A user with administrator privileges must be logged in to perform this operation.

User.AddUserGroup Method

Creates new user group.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddUserGroup", Description = "")]
public void AddUserGroup(string GroupName, string GroupPath, string GroupDomain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddUserGroup", Description = "")>
Public Function AddUserGroup(GroupName As string, GroupPath As string, GroupDomain As string)
As void
```

Parameters

Parameters	Description
string GroupName	The name of the group.
string GroupPath	Group path, usually required for Active Directory (see page 196), otherwise CMS group assumed.
string GroupDomain	Group domain, usually required for Active Directory (see page 196).

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.AddUserToGroup Method

Adds an existing user to an existing group.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddUserToGroup", Description = "")]
public void AddUserToGroup(long UserId, long GroupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddUserToGroup", Description = "")>
Public Function AddUserToGroup(UserId As long, GroupId As long) As void
```

Parameters

Parameters	Description
long UserId	The user's ID.
long GroupId	The required group's ID.

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.AutoLogInUser Method

Login (see page 347) the given user credentials.

This method can be accessed through the User.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/User/User.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AutoLogInUser", Description = "")]
public UserData AutoLogInUser(string SAMusername, string domain, string ServerName);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AutoLogInUser", Description = "")>
Public Function AutoLogInUser(SAMusername As string, domain As string, ServerName As string)
As UserData
```

Parameters

Parameters	Description
string SAMusername	The user's name.
string domain	The domain.
string ServerName	The server name.

CreateBlankUserObject Method

This is the overview for the CreateBlankUserObject method overload.

Description

This method is overloaded.

- CreateBlankUserObject () - returns the instance of the user data.
- CreateBlankUserObject (ref UserData) - sets the given user data values into the UserData class.

Overload List

	Name	Description
	User.CreateBlankUserObject () (see page 1376)	Returns the instance of the user data. This function is overloaded. See Also: CreateBlankUserObject(UserData) This method can be accessed through the User.asmx file located in:
	User.CreateBlankUserObject (UserData) (see page 1377)	Sets the given user data values into the UserData class. This function is overloaded. See Also: CreateBlankUserObject This method can be accessed through the User.asmx file located in:

User.CreateBlankUserObject Method ()

Returns the instance of the user data.

This function is overloaded. See Also: CreateBlankUserObject(UserData)

This method can be accessed through the User.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/User/User.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CreateBlankUserObject", Description = "")]
public UserData CreateBlankUserObject();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CreateBlankUserObject", Description = "")>
Public Function CreateBlankUserObject() As UserData
```

Returns

Ektron.Cms.UserData

User.CreateBlankUserObject Method (UserData)

Sets the given user data values into the UserData class.

This function is overloaded. See Also: CreateBlankUserObject

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CreateBlankUserObject By UserData", Description = "")]
public void CreateBlankUserObject(ref UserData UserData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CreateBlankUserObject By UserData", Description = "")>
Public Function CreateBlankUserObject(ByRef UserData As UserData) As void
```

Parameters

Parameters	Description
ref UserData UserData	an Object of user properties

User.DeleteUserByID Method

Removes the user from the system. This will not purge the data, but it disables the user from further accessing the system.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteUserByID", Description = "")]
public void DeleteUserByID(long UserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteUserByID", Description = "")>
Public Function DeleteUserByID(UserId As long) As void
```

Parameters

Parameters	Description
long UserId	The user's ID.

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.DeleteUserByIds Method

Removes the user from the system. This will not purge the data, but it disables the user from further accessing the system.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteUserByIds", Description = "")]
public void DeleteUserByIds(string Ids);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteUserByIds", Description = "")>
Public Function DeleteUserByIds(Ids As string) As void
```

Parameters

Parameters	Description
string Ids	The selected user IDs(comma separated values).

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.DeleteUserFromGroup Method

Removes a user from the designated group.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteUserFromGroup", Description = "")]
public void DeleteUserFromGroup(long UserId, long GroupId, bool ReturnDeleted);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteUserFromGroup", Description = "")>
Public Function DeleteUserFromGroup(UserId As long, GroupId As long, ReturnDeleted As bool) As
void
```

Parameters

Parameters	Description
long UserId	The user's ID.
long GroupId	The group's ID.

bool ReturnDeleted	

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.DeleteUserGroup Method

Removes the group from the system permanently.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteUserGroup", Description = "")]
public void DeleteUserGroup(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteUserGroup", Description = "")>
Public Function DeleteUserGroup(Id As long) As void
```

Parameters

Parameters	Description
long Id	The group's ID.

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.EditUserCustomProperties Method

This method creates the editable fields(UI) for user's custom properties.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "EditUserCustomProperties", Description = "")]
public string EditUserCustomProperties(long UserId, bool ReadOnly);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "EditUserCustomProperties", Description = "")>
Public Function EditUserCustomProperties(UserId As long, ReadOnly As bool) As string
```

Parameters

Parameters	Description
long UserId	User (see page 1369)'s ID. If the user ID is not specified then it will return empty fields.
bool ReadOnly	

Returns

String

User.GetActiveUser Method

Returns all of the user properties with or without preference.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetActiveUser", Description = "")]
public UserData GetActiveUser(long Id, bool Preference);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetActiveUser", Description = "")>
Public Function GetActiveUser(Id As long, Preference As bool) As UserData
```

Parameters

Parameters	Description
long Id	The user's ID.
bool Preference	Preference is returned only if the value true, otherwise nothing is returned for preference.

Returns

Ektron.Cms.UserData

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetADMapping Method

Loads the CMS and Active Directory (see page 196) name.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetADMapping", Description = "")]
public AdMappingData[] GetADMapping(long CallerId, string MapType, int Map, int Exclude, int
Editable);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetADMapping", Description = "")>
Public Function GetADMapping(CallerId As long, MapType As string, Map As Integer, Exclude As
Integer, Editable As Integer) As AdMappingData()
```

Parameters

Parameters	Description
long CallerId	Who is calling this function.
string MapType	Possible value is “userprop” (user properties).
int Map	Ektron.Cms.Common.EkEnumeration.MapTypes (include, exclude and dontconsider are the possible values).
int Exclude	Flag (see page 1231) (0 – false, 1-true).
int Editable	Flag (see page 1231)(0-false,1-true).

Returns

Ektron.Cms.AdMappingData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetAllCmsUsers Method

Loads all the CMS users.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllCmsUsers", Description = "")]
public UserData[] GetAllCmsUsers(string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllCmsUsers", Description = "")>
Public Function GetAllCmsUsers(OrderBy As string) As UserData()
```

Parameters

Parameters	Description
string OrderBy	Used to arrange data in a specific order. The string value can be: • username • firstname • lastname • userid • language • datecreated • lastlogindate

Returns

Ektron.Cms.UserData()

Description

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetAllCustomProperties Method

Returns array of all the custom properties defined in the system.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllCustomProperties", Description = "")]
public CustomAttribute[] GetAllCustomProperties();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllCustomProperties", Description = "")>
Public Function GetAllCustomProperties() As CustomAttribute()
```

Returns

Ektron.Cms.CustomAttribute()

User.GetAllUserByCustomProperty Method

Returns all the users based on the given properties.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUserByCustomProperty", Description = "")]
public UserData[]
GetAllUserByCustomProperty(Ektron.Cms.Common.EkEnumeration.ObjectPropertyValueTypes PropType,
string PropValues);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUserByCustomProperty", Description = "")>
Public Function GetAllUserByCustomProperty(PropType As
Ektron.Cms.Common.EkEnumeration.ObjectPropertyValueTypes, PropValues As string) As UserData()
```

Parameters

Parameters	Description
Ektron.Cms.Common.EkEnumeration.ObjectPropertyValueTypes PropType	
string PropValues	Required property

Remarks

A user with administrator privileges must be logged in to perform this operation.

GetAllUserGroups Method

This is the overview for the GetAllUserGroups method overload.

Description

This method is overloaded.

- GetAllUserGroups (EkEnumeration.UserTypes, string) (see page 1383) - loads all of the user groups from the system based on group type and order by information.
- GetAllUserGroups (string) (see page 1383) - loads all of the user groups from the system and allows you to specify order by information.

Overload List

	Name	Description
	User.GetAllUserGroups (EkEnumeration.UserTypes, string) (see page 1383)	Loads all of the user groups from the system. This function is overloaded. See Also: GetAllUserGroups(String) This method can be accessed through the User.asmx file located in:
	User.GetAllUserGroups (string) (see page 1383)	Loads all of the user groups from the system. This function is overloaded. See Also: GetAllUserGroups(EkEnumeration.UserTypes, String) This method can be accessed through the User.asmx file located in:

User.GetAllUserGroups Method (EkEnumeration.UserTypes, string)

Loads all of the user groups from the system.

This function is overloaded. See Also: GetAllUserGroups(String)

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUserGroups By GroupType and OrderBy", Description = "")]
public UserGroupData[] GetAllUserGroups(EkEnumeration.UserTypes GroupType, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUserGroups By GroupType and OrderBy", Description = "")>
Public Function GetAllUserGroups(GroupType As EkEnumeration.UserTypes, OrderBy As string) As
UserGroupData()
```

Parameters

Parameters	Description
EkEnumeration.UserTypes GroupType	GroupType: 1 = Membership (see page 360); 0 = CMSUser
string OrderBy	Arranges data in a specific order. Possible values are GroupName or GroupID.

Returns

Ektron.Cms.UserGroupData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetAllUserGroups Method (string)

Loads all of the user groups from the system.

This function is overloaded. See Also: GetAllUserGroups(EkEnumeration.UserTypes, String)

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "GetAllUserGroups", Description = "")]
public UserGroupData[] GetAllUserGroups(string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUserGroups", Description = "")>
Public Function GetAllUserGroups(OrderBy As string) As UserGroupData()
```

Parameters

Parameters	Description
string OrderBy	Arranges data in a specific order. Possible values are GroupName or GroupID.

Returns

Ektron.Cms.UserGroupData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

GetAllUsers Method

This is the overview for the GetAllUsers method overload.

Description

This method is overloaded.

GetAllUsers (Long, Long) - gets the User Group Data of all the allowed users in the folder specified.

GetAllUsers (ref UserRequestData) - returns the list of users from the database based on the UserRequestData class.

Overload List

	Name	Description
	User.GetAllUsers (long, long) (see page 1385)	Gets the User (see page 1369) Group Data of all the allowed users in the folder specified. This method can be accessed through the User.asmx file located in:
	User.GetAllUsers (UserRequestData) (see page 1384)	Returns the list of users from the database based on the UserRequestData class. This method can be accessed through the User.asmx file located in:

User.GetAllUsers Method (UserRequestData)

Returns the list of users from the database based on the UserRequestData class.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUsers by UserRequest", Description = "")]
public UserData[] GetAllUsers(ref UserRequestData userRequest);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUsers by UserRequest", Description = "")>
Public Function GetAllUsers(ByRef userRequest As UserRequestData) As UserData()
```

Parameters

Parameters	Description
ref UserRequestData userRequest	list of constraints to load users

Returns

An array of users as UserData.

User.GetAllUsers Method (long, long)

Gets the User (see page 1369) Group Data of all the allowed users in the folder specified.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUsers By CallerId and FolderID", Description = "")]
public UserGroupData[] GetAllUsers(long CallerID, long FolderId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUsers By CallerId and FolderID", Description = "")>
Public Function GetAllUsers(CallerID As long, FolderId As long) As UserGroupData()
```

Parameters

Parameters	Description
long CallerID	The user ID that is logged on to the system.
long FolderId	The ID of the folder.

Returns

Ektron.Cms.UserGroupData()

Remarks

User (see page 1369) login with administrator privileges are required to perform this operation.

User.GetGroupsUserIsIn Method

Loads all the groups by user.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetGroupsUserIsIn", Description = "")]
public GroupData[] GetGroupsUserIsIn(long UserId, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetGroupsUserIsIn", Description = "")>
Public Function GetGroupsUserIsIn(UserId As long, OrderBy As string) As GroupData()
```

Parameters

Parameters	Description
long UserId	The user's ID.
string OrderBy	The order in which the groups are displayed. The possible values are "GroupName" and "GroupID". (Required)

Returns

Ektron.Cms.GroupData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetMembershipUser Method

Returns user data based on the user's login name (username).

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMembershipUser", Description = "")]
public UserData GetMembershipUser(string Username);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMembershipUser", Description = "")>
Public Function GetMembershipUser(Username As string) As UserData
```

Parameters

Parameters	Description
string Username	User (see page 1369)'s login name (username). This is required

Returns

Ektron.Cms.UserData

Remarks

User (see page 1369) login with administrator privileges are required to perform this operation.

User.GetUser Method

Loads the user details for the given user.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUser", Description = "")]
public UserData GetUser(long Id, bool Preference, bool ReturnDeleted);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUser", Description = "")>
```

```
Public Function GetUser(Id As long, Preference As bool, ReturnDeleted As bool) As UserData
```

Parameters

Parameters	Description
long Id	The user ID.
bool Preference	Loads preference only if required (true = load preferences as a part of user data).

Returns

Ektron.Cms.UserData

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetUserbyUsername Method

Loads the user details for the given user.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserbyUsername", Description = "")]
public UserData GetUserbyUsername(string username);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserbyUsername", Description = "")>
Public Function GetUserbyUsername(username As string) As UserData
```

Parameters

Parameters	Description
string username	The user username.

Returns

Ektron.Cms.UserData

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetUserCustomProperties Method

Returns an array of custom properties for a given user's ID.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserCustomProperties", Description = "")]
public CustomAttribute[] GetUserCustomProperties(long UserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "GetUserCustomProperties", Description = "")>
Public Function GetUserCustomProperties(UserId As long) As CustomAttribute()
```

Parameters

Parameters	Description
long UserId	User (see page 1369)'s ID

Returns

Ektron.Cms.CustomAttribute()

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetUserEmailInfo Method

Returns user data for a given user's ID.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserEmailInfo", Description = "")]
public UserData GetUserEmailInfo(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserEmailInfo", Description = "")>
Public Function GetUserEmailInfo(Id As long) As UserData
```

Parameters

Parameters	Description
long Id	The user's ID.

Returns

Object of UserData

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetUserForFolderAdmin Method

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserForFolderAdmin", Description = "")]
public UserData GetUserForFolderAdmin(long FolderId, long Id, bool Preference, bool
ReturnDeleted);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserForFolderAdmin", Description = "")>
Public Function GetUserForFolderAdmin(FolderId As long, Id As long, Preference As bool,
ReturnDeleted As bool) As UserData
```

Parameters

Parameters	Description
long FolderId	The ID of the folder.
long Id	
bool Preference	
bool ReturnDeleted	

User.GetUserGroup Method

Loads a given group's details.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserGroup", Description = "")]
public UserGroupData GetUserGroup(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserGroup", Description = "")>
Public Function GetUserGroup(Id As long) As UserGroupData
```

Parameters

Parameters	Description
long Id	The group's ID.

Returns

Ektron.Cms.UserGroupData

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetUserGroupForFolderAdmin Method

Returns all the groups data for the administrator.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserGroupForFolderAdmin", Description = "")]
public UserGroupData GetUserGroupForFolderAdmin(long FolderId, long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserGroupForFolderAdmin", Description = "")>
Public Function GetUserGroupForFolderAdmin(FolderId As long, Id As long) As UserGroupData
```

Parameters

Parameters	Description
long FolderId	The ID of the folder.
long Id	The ID of the group.

Returns

UserGroupData

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetUserPermissions Method

Loads the user permissions as a array of UserPermissionData.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserPermissions", Description = "")]
public UserPermissionData[] GetUserPermissions(long Id, string ItemType, long UserId, string
UserGroupList, EkContent.PermissionUserType PermissionType, EkContent.PermissionRequestType
PermissionRequest);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserPermissions", Description = "")>
Public Function GetUserPermissions(Id As long, ItemType As string, UserId As long,
UserGroupList As string, PermissionType As EkContent.PermissionUserType, PermissionRequest As
EkContent.PermissionRequestType) As UserPermissionData()
```

Parameters

Parameters	Description
long Id	The item's numeric ID.
string ItemType	The item's type.
long UserId	The user's numeric ID.
string UserGroupList	The group list.
EkContent.PermissionUserType PermissionType	PermissionUserType
EkContent.PermissionRequestType PermissionRequest	PermissionRequestType

Returns

Ektron.Cms.UserPermissionData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetUserPreference Method

Loads the preferences for the given user.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserPreference", Description = "")]
public UserPreferenceData GetUserPreference(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserPreference", Description = "")>
Public Function GetUserPreference(Id As long) As UserPreferenceData
```

Parameters

Parameters	Description
long Id	The user's ID.

Returns

Ektron.Cms.UserPreferenceData

Remarks

A user with administrator privileges must be logged in to perform this operation.

GetUsers Method

This is the overview for the GetUsers method overload.

Description

This method is overloaded.

- GetUsers (Long, String) - get the users from a specific group or load and array of user group data.
- GetUser (String, String) - returns an array of user group data based on the userName and userNameType.

Overload List

	Name	Description
	User.GetUsers (long, string) (see page 1391)	Loads the all the users from the given group. This method can be accessed through the User.asmx file located in:
	User.GetUsers (string, string) (see page 1392)	Returns an array of user group data based on the userName and userNameType. This method can be accessed through the User.asmx file located in:

User.GetUsers Method (long, string)

Loads the all the users from the given group.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUsers By GroupID and OrderBy", Description = "")]
public UserData[] GetUsers(long GroupId, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUsers By GroupID and OrderBy", Description = "")>
Public Function GetUsers(GroupId As long, OrderBy As string) As UserData()
```

Parameters

Parameters	Description
long GroupId	The group's ID
string OrderBy	The order of the users. The string value can be: • username • firstname • lastname • userid • language • datecreated • lastlogindate

Returns

Ektron.Cms.UserData()

Description

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetUsers Method (string, string)

Returns an array of user group data based on the userName and userNameType.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUsers", Description = "")]
public UserGroupData[] GetUsers(string userNames, string userNameTypes);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUsers", Description = "")>
Public Function GetUsers(userNames As string, userNameTypes As string) As UserGroupData()
```

Parameters

Parameters	Description
string userNames	The user's name. (Required)
string userNameTypes	The user type possible values are user or membership user.

Returns

Ektron.Cms.UserGroupData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetUsersForTask Method

Loads the users that have permissions to the given content.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUsersForTask", Description = "")]
public UserGroupData[] GetUsersForTask(long CallerID, long ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUsersForTask", Description = "")>
Public Function GetUsersForTask(CallerID As long, ContentID As long) As UserGroupData()
```

Parameters

Parameters	Description
long CallerID	The user ID that is logged on to the system.
long ContentID	The ID of the content. (Required)

Returns

Ektron.Cms.UserGroupData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.GetUsersNotInGroup Method

Loads all the users that are not in the given group.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUsersNotInGroup", Description = "")]
public UserData[] GetUsersNotInGroup(long Id, string OrderBy, int GroupType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUsersNotInGroup", Description = "")>
Public Function GetUsersNotInGroup(Id As long, OrderBy As string, GroupType As Integer) As
UserData()
```

Parameters

Parameters	Description
long Id	The group ID.

string OrderBy	The order of the users. The string value can be: • username • firstname • lastname • userid • language • datecreated • lastlogindate

Returns

Ektron.Cms.UserData()

Description

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.IsAdmin Method

Determines the if the current user is an Administrator.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsAdmin", Description = "")]
public bool IsAdmin();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsAdmin", Description = "")>
Public Function IsAdmin() As bool
```

Returns

True if the logged in user is administrator. Otherwise, returns false.

IsARoleMember Method

This is the overview for the IsARoleMember method overload.

Description

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This method is overloaded.

Overload List

	Name	Description
	User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) (see page 1396)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This function is overloaded. See Also: • CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean) • CommonApi.IsARoleMember Method (Integer, Integer, Boolean) • CommonApi.IsARoleMember Method (Integer) This method can be accessed through the User.asmx file located in:
	User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, long, bool) (see page 1396)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This function is overloaded. See Also: • CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) • CommonApi.IsARoleMember Method (Integer, Integer, Boolean) • CommonApi.IsARoleMember Method (Integer) This method can be accessed through the User.asmx file located in:
	User.IsARoleMember (long) (see page 1397)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This function is overloaded. See Also: • CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) • CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean) • CommonApi.IsARoleMember Method (Integer, Integer, Boolean) This method can be accessed through the User.asmx file located in:

	User.IsARoleMember (long, long, bool) (see page 1397)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This function is overloaded. See Also: • CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean) • CommonApi.IsARoleMember Method (Integer) • CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) This method can be accessed through the User.asmx file located in:

User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean)
- CommonApi.IsARoleMember Method (Integer, Integer, Boolean)
- CommonApi.IsARoleMember Method (Integer)

This method can be accessed through the User.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/User/User.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMember by CMSroleID", Description = "")]
public bool IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMember by CMSroleID", Description = "")>
Public Function IsARoleMember(RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds) As bool
```

Parameters

Parameters	Description
Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId	_nt_

User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, long, bool)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)
- CommonApi.IsARoleMember Method (Integer, Integer, Boolean)
- CommonApi.IsARoleMember Method (Integer)

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMember by CmsRoleID, UserID and GroupFlag", Description = "")]
public bool IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId, long UserId, bool
GroupFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMember by CmsRoleID, UserID and GroupFlag", Description = "")>
Public Function IsARoleMember(RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds, UserId As
long, GroupFlag As bool) As bool
```

Parameters

Parameters	Description
Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId	_nt_
bool GroupFlag	_nt_
nUserId	_nt_

User.IsARoleMember Method (long)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)
- CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean)
- CommonApi.IsARoleMember Method (Integer, Integer, Boolean)

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMember by RoleID", Description = "")]
public bool IsARoleMember(long RoleId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMember by RoleID", Description = "")>
Public Function IsARoleMember(RoleId As long) As bool
```

Parameters

Parameters	Description
long RoleId	_nt_

User.IsARoleMember Method (long, long, bool)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean)
- CommonApi.IsARoleMember Method (Integer)
- CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)

This method can be accessed through the User.asmx file located in:

~/Workarea/webservices/WebServiceAPI/User/User.asmx

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMember by RoleID, UserId and GroupFlag", Description = "")]
public bool IsARoleMember(long RoleId, long UserId, bool GroupFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMember by RoleID, UserId and GroupFlag", Description = "")>
Public Function IsARoleMember(RoleId As long, UserId As long, GroupFlag As bool) As bool
```

Parameters

Parameters	Description
long RoleId	_nt_
bool GroupFlag	_nt_
nUserId	_nt_

User.IsARoleMemberForContent Method

Determines whether a Role Member is allowed to work with a given piece of content as an administrator or not. Administrations privileges are for the given piece of content only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This method can be accessed through the User.asmx file located in:

~/Workarea/webservices/WebServiceAPI/User/User.asmx

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMemberForContent", Description = "")]
public bool IsARoleMemberForContent(long RoleId, long ContentId, long UserId, bool GroupFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMemberForContent", Description = "")>
Public Function IsARoleMemberForContent(RoleId As long, ContentId As long, UserId As long,
GroupFlag As bool) As bool
```

Parameters

Parameters	Description
long RoleId	
long ContentId	The numeric content ID for which you are giving the Role Member privileges.
bool GroupFlag	
nUserId	

User.IsARoleMemberForFolder Method

Determines whether a Role Member is allowed to work with a given folder as an administrator or not. Administrations privileges are for the given folder only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users,

without giving full administration privileges to a user.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMemberForFolder", Description = "")]
public bool IsARoleMemberForFolder(long RoleId, long FolderId, long UserId, bool GroupFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMemberForFolder", Description = "")>
Public Function IsARoleMemberForFolder(RoleId As long, FolderId As long, UserId As long,
GroupFlag As bool) As bool
```

Parameters

Parameters	Description
long RoleId	
long FolderId	The numeric folder ID for which you are giving the Role Member privileges.
long UserId	
bool GroupFlag	

User.LogInUser Method

Used to log in the user.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "LogInUser", Description = "")]
public UserData LogInUser(string username, string Password, string ServerName, string Domain,
string Protocol);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "LogInUser", Description = "")>
Public Function LogInUser(username As string, Password As string, ServerName As string, Domain
As string, Protocol As string) As UserData
```

Parameters

Parameters	Description
string username	The system username.
string Password	The system password.
string ServerName	The system server name.
string Domain	The user's domain, required only when users login using Active Directory (see page 196).
string Protocol	Required only when using a secure login or when you are using any valid Active Directory (see page 196) protocols.

Returns

Ektron.Cms.UserData

ResetMembershipUserPassword Method

This is the overview for the ResetMembershipUserPassword method overload.

Description

This method is overloaded.

- ResetMembershipUserPassword (string) (see page 1400) - reset the password for the given membership user by providing the username.
- ResetMembershipUserPassword (string, string) (see page 1400) - reset the password for the given membership user by providing the username and verification account ID.

Overload List

	Name	Description
	User.ResetMembershipUserPassword (string) (see page 1400)	Reset the password for the given membership user. This method can be accessed through the User.asmx file located in:
	User.ResetMembershipUserPassword (string, string) (see page 1400)	Reset the password for the given membership user. This method can be accessed through the User.asmx file located in:

User.ResetMembershipUserPassword Method (string)

Reset the password for the given membership user.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ResetMembershipUserPassword", Description = "")]
public string ResetMembershipUserPassword(string Username);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ResetMembershipUserPassword", Description = "")>
Public Function ResetMembershipUserPassword(Username As string) As string
```

Parameters

Parameters	Description
string Username	User (see page 1369)'s username to be reset

Returns

String

Remarks

User (see page 1369) login is required to perform this operation.

User.ResetMembershipUserPassword Method (string, string)

Reset the password for the given membership user.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ResetMembershipUserPasswordWithVerify", Description = "")]
public string ResetMembershipUserPassword(string Username, string AccountId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ResetMembershipUserPasswordWithVerify", Description = "")>
Public Function ResetMembershipUserPassword(Username As string, AccountId As string) As string
```

Parameters

Parameters	Description
string Username	User (see page 1369)'s username to be reset
string AccountId	If verification is turned on, specify verification id(email to a user)

Returns

String

Remarks

User (see page 1369) login is required to perform this operation.

User.UnsubscribeUser Method

Unsubscribes a user from the system.

This method can be accessed through the User.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/User/User.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UnsubscribeUser", Description = "")]
public bool UnsubscribeUser(string Username, string Password);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UnsubscribeUser", Description = "")>
Public Function UnsubscribeUser(Username As string, Password As string) As bool
```

Parameters

Parameters	Description
string Username	The user's username
string Password	The user's password

Returns

Boolean

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.UpdateCustomPropertiesItemOrder Method

Updates the order of available custom properties.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateCustomPropertiesItemOrder", Description = "")]
public bool UpdateCustomPropertiesItemOrder(string ItemOrder);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateCustomPropertiesItemOrder", Description = "")>
Public Function UpdateCustomPropertiesItemOrder(ItemOrder As string) As bool
```

Parameters

Parameters	Description
string ItemOrder	Order of the properties applicable to the user.

Returns

Boolean

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.UpdateMapCoordinate Method

Sets the map co-ordinate for the given user property.

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateMapCoordinate", Description = "")]
public void UpdateMapCoordinate(long userId, double latitude, double longitude);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateMapCoordinate", Description = "")>
Public Function UpdateMapCoordinate(userId As long, latitude As double, longitude As double)
As void
```

Parameters

Parameters	Description
long userId	The user's ID.
attribId	Map (see page 350) latitude/longitude numeric ID.
value	Map (see page 350) latitude or longitude value.

Remarks

A user with administrator privileges must be logged in to perform this operation.

UpdateUser Method

This is the overview for the UpdateUser method overload.

Description

This method is overloaded.

- UpdateUser (Collection) (see page 1403) - updates the user details based on a collection.
- UpdateUser (UserData) (see page 1403) - updates the user details based on provided UserData.

Overload List

	Name	Description
	User.UpdateUser (Collection) (see page 1403)	Updates the user details. This method can be accessed through the User.asmx file located in:
	User.UpdateUser (UserData) (see page 1403)	Updates the user details. This method can be accessed through the User.asmx file located in:

User.UpdateUser Method (Collection)

Updates the user details.

This method can be accessed through the User.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/User/User.asmx`

C#

```
public void UpdateUser(Collection data);
```

Visual Basic

```
Public Function UpdateUser(data As Collection) As void
```

Parameters

Parameters	Description
Collection data	A collection of user data.

Description

[WebMethod(MessageName = "UpdateUser", Description = "")]

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.UpdateUser Method (UserData)

Updates the user details.

This method can be accessed through the User.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/User/User.asmx`

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateUser", Description = "")]
public void UpdateUser(UserData data);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateUser", Description = "")>
Public Function UpdateUser(data As UserData) As void
```

Parameters

Parameters	Description
UserData data	A class of type UserData.

Remarks

User (see page 1369) login with administrator privileges are required to perform this operation.

User.UpDateUserGroup Method

Updates the user group details.

This method can be accessed through the User.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/User/User.asmx`

C#

```
public void UpDateUserGroup(UserData data);
```

Visual Basic

```
Public Function UpDateUserGroup(data As UserData) As void
```

Parameters

Parameters	Description
UserData data	A class of type UserData.

Description

[WebMethod(MessageName = "UpDateUserGroup", Description = "")]

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.UpdateUserPreferences Method

Updates the given user's preference data.

This method can be accessed through the User.asmx file located in:

`~/Workarea/webservices/WebServiceAPI/User/User.asmx`

C#

```
public void UpdateUserPreferences(long Id, Collection data);
```

Visual Basic

```
Public Function UpdateUserPreferences(Id As long, data As Collection) As void
```

Parameters

Parameters	Description
long Id	The user's ID.
Collection data	A collection of user preferences.

Description

[WebMethod(MessageName = "UpdateUserPreferences", Description = "")]

Remarks

A user with administrator privileges must be logged in to perform this operation.

UserObject Method

This is the overview for the UserObject method overload.

Description

This method is overloaded.

- UserObject (Long)- returns user data based on the user's ID.
- UserObject (String) - returns user data based on the user's login or username.

Overload List

	Name	Description
	User.UserObject (long) (see page 1405)	Returns user data based on the user's ID. This function is overloaded. See Also: UserObject(String) This method can be accessed through the User.asmx file located in:
	User.UserObject (string) (see page 1405)	Returns user data based on the user's login or username. This function is overloaded. See Also: UserObject(Long) This method can be accessed through the User.asmx file located in:

User.UserObject Method (long)

Returns user data based on the user's ID.

This function is overloaded. See Also: UserObject(String)

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UserObject By User ID", Description = "")]
public UserData UserObject(long Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UserObject By User ID", Description = "")>
Public Function UserObject(Id As long) As UserData
```

Parameters

Parameters	Description
long Id	The user's ID.

Returns

Ektron.Cms.UserData

User.UserObject Method (string)

Returns user data based on the user's login or username.

This function is overloaded. See Also: UserObject(Long)

This method can be accessed through the User.asmx file located in:

```
~/Workarea/webservices/WebServiceAPI/User/User.asmx
```

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UserObject By UserName", Description = "")]
public UserData UserObject(string UserName);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UserObject By UserName", Description = "")>
Public Function UserObject(UserName As string) As UserData
```

Parameters

Parameters	Description
sUserName	User (see page 1369)'s login or username.

Returns

Ektron.Cms.UserData

Providers API

The Providers API section describes the methods and properties needed to create custom payment gateways, inventory and shipping providers in code behind for use with CMS400.NET.

Payment Gateway Providers

A Payment Gateway Provider is a pluggable component that is integrated into Ektron's CMS400.NET eCommerce (see page 601) module. A Payment provider handles eCommerce (see page 601) customer payments by utilizing third party payment gateways. CMS400.NET's eCommerce (see page 601) module accepts payments such as, credit cards. Then, it passes that information to a third party service. The third party service processes the payment, and returns a transaction ID that's stored with the customer's order.

Classes

	Name	Description
	PaymentGatewayProvider (see page 1412)	Base class for Payment Gateway Providers. A Payment Gateway Provider (see page 1407) is a pluggable component that is integrated into Ektron's CMS400.NET eCommerce (see page 601) module. A Payment provider handles eCommerce (see page 601) customer payments by utilizing third party payment gateways. CMS400.NET's eCommerce (see page 601) module accepts payments such as, credit cards. Then, it passes that information to a third party service. The third party service processes the payment, and returns a transaction ID that's stored with the customer's order. You can use this base class to create your own custom payment gateway. For additional information on payment gateways, see the Ektron CMS400.NET Developer Manual's "Customizing eCommerce (see page 601)... more (see page 1412)

Namespaces

Name	Description
Inventory (see page 1407)	This is namespace Ektron.Cms.Commerce.Inventory.
Shipment (see page 1418)	This is namespace Ektron.Cms.Commerce.Shipment.

Ektron.Cms.Commerce.Inventory Namespace

This is namespace Ektron.Cms.Commerce.Inventory.

Namespaces

Name	Description
Provider (see page 1407)	This is namespace Ektron.Cms.Commerce.Inventory.Provider.

Ektron.Cms.Commerce.Inventory.Provider Namespace

This is namespace Ektron.Cms.Commerce.Inventory.Provider.

Classes

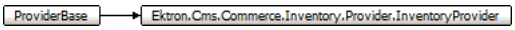
	Name	Description
	InventoryProvider (see page 1407)	This is the base class for the inventory provider. Use this class to create a custom inventory provider. When creating a custom inventory provider using this class, you must implement the abstract methods: GetInventory (see page 1409) and SaveInventory (see page 1411). The other methods are optional and can be overridden as needed. See the Ektron CMS400.NET Developer Manual's "Customizing eCommerce (see page 601)" > "Inventory (see page 1407) Provider" section for more information and a step-by-step example of creating a custom inventory provider.

InventoryProvider Class

This is the base class for the inventory provider. Use this class to create a custom inventory provider. When creating a custom inventory provider using this class, you must implement the abstract methods: GetInventory (see page 1409) and SaveInventory (see page 1411). The other methods are optional and can be overridden as needed.

See the Ektron CMS400.NET Developer Manual's "Customizing eCommerce (see page 601)" > "Inventory (see page 1407) Provider (see page 1407)" section for more information and a step-by-step example of creating a custom inventory provider.

Class Hierarchy



C#

```
public abstract class InventoryProvider : ProviderBase;
```

Visual Basic

```
Public abstract Class InventoryProvider
Inherits ProviderBase
```

File

InventoryProvider.cs

InventoryProvider Methods

The methods of the InventoryProvider class are listed here.

Description

This section contains methods for the InventoryProvider class.

Public Methods

	Name	Description
	DecreaseStockLevel (see page 1408)	Decreases the stock level for a catalog entry (product) based on its ID and the given quantity. For example, if a site visitor purchases two copies of the same book (ID=12) from your site, you might use the following line of code to decrease the book's in-stock quantity by two:
	GetInventory (see page 1409)	Gets inventory information for a specified catalog entry (product). Some of the items returned include: • EntryId • UnitsInStock • UnitsOnOrder • ReorderLevel For example, if you want to find out the number of units in stock, units on order, and the re-order level for an item with an ID of 12, you might use the following line of code:
	GetReorderLevel (see page 1409)	Returns the re-order level for a given catalog entry (product). The re-order level is the inventory level at which new stock should be ordered. Use this method when you only want to return the re-order level for a specified item. If you want additional inventory information about an item, use the GetInventory (see page 1409) method. Below is an example of using this method in code to check the re-order level for an item with an ID of 12.
	GetUnitsInStock (see page 1410)	Gets the number of units in stock for a given catalog entry's ID. Use this method when you only want to return the number of units in stock for a specified item. If you want additional inventory information about an item, use the GetInventory (see page 1409) method. Below is example code that returns the number of units in stock for an item with an ID of 12.
	GetUnitsOnOrder (see page 1410)	Gets the number of units on order for a given catalog entry's ID. This is the number of units that the store has ordered to replenish inventory and NOT the number of units that are currently on order by customers. Use this method when you only want to return the number of units on order for a specified item. If you want additional inventory information about an item, use the GetInventory (see page 1409) method. Below is example code that returns the number of units on order for an item with an ID of 12.
	IncreaseStockLevel (see page 1410)	Increases the stock level for a catalog entry (product) based on its ID and the given quantity. For example, if you need to increase the stock level of a given item (for example, item ID=12) by ten in your inventory system, you might use the following line of code:
	SaveInventory (see page 1411)	Saves inventory information for a catalog entry. Some of the items saved include: • EntryId • UnitsInStock • UnitsOnOrder • ReorderLevel For example, if you have an item with the ID of 12 and you want to set its units in stock to 3, the units on order to 10, and the re-order level 5, you might use the following lines of code:

InventoryProvider.DecreaseStockLevel Method

Decreases the stock level for a catalog entry (product) based on its ID and the given quantity. For example, if a site visitor purchases two copies of the same book (ID=12) from your site, you might use the following line of code to decrease the book's in-stock quantity by two:

```
inventory.DecreaseStockLevel(12, 2);
```

C#

```
public virtual void DecreaseStockLevel(long entryId, int quantity);
```

Visual Basic

```
Public virtual Function DecreaseStockLevel(entryId As long, quantity As Integer) As void
```

Parameters

Parameters	Description
long entryId	Enter the numeric ID of the catalog entry (product).

int quantity	Enter the numeric quantity by which the stock level will decrease.

Remarks

The default implementation uses GetInventory (see page 1409)() and SaveInventory (see page 1411)().

InventoryProvider.GetInventory Method

Gets inventory information for a specified catalog entry (product). Some of the items returned include:

- EntryId
- UnitsInStock
- UnitsOnOrder
- ReorderLevel

For example, if you want to find out the number of units in stock, units on order, and the re-order level for an item with an ID of 12, you might use the following line of code:

```
inventory.GetInventory(12);
```

When creating a custom inventory provider you must implement this method.

If you only want to return the re-order level, you could also use the GetReorderLevel (see page 1409) method.

If you only want to return the number of units in stock, you could also use the GetUnitsInStock (see page 1410) method.

If you only want to return the number of units on order, you could also use the GetUnitsOnOrder (see page 1410) method.

C#

```
public abstract InventoryData GetInventory(long entryId);
```

Visual Basic

```
Public abstract Function GetInventory(entryId As long) As InventoryData
```

Parameters

Parameters	Description
long entryId	Enter the numeric ID for a catalog entry (product).

Returns

Returns the InventoryData (see page 1506) data class object for the supplied catalog entry.

InventoryProvider.GetReorderLevel Method

Returns the re-order level for a given catalog entry (product). The re-order level is the inventory level at which new stock should be ordered. Use this method when you only want to return the re-order level for a specified item. If you want additional inventory information about an item, use the GetInventory (see page 1409) method. Below is an example of using this method in code to check the re-order level for an item with an ID of 12.

```
inventory.GetReorderLevel(12);
```

C#

```
public virtual int GetReorderLevel(long entryId);
```

Visual Basic

```
Public virtual Function GetReorderLevel(entryId As long) As Integer
```

Parameters

Parameters	Description
long entryId	Enter the numeric ID of the catalog entry.

Returns

Returns the level at which an item should be re-ordered.

Remarks

The default implementation uses GetInventory (see page 1409)().

InventoryProvider.GetUnitsInStock Method

Gets the number of units in stock for a given catalog entry's ID. Use this method when you only want to return the number of units in stock for a specified item. If you want additional inventory information about an item, use the GetInventory (see page 1409) method. Below is example code that returns the number of units in stock for an item with an ID of 12.

```
inventory.GetUnitsInStock(12);
```

C#

```
public virtual int GetUnitsInStock(long entryId);
```

Visual Basic

```
Public virtual Function GetUnitsInStock(entryId As long) As Integer
```

Parameters

Parameters	Description
long entryId	Enter the ID for a catalog entry (product).

Returns

Returns the number of units in stock for a given catalog entry's ID.

InventoryProvider.GetUnitsOnOrder Method

Gets the number of units on order for a given catalog entry's ID. This is the number of units that the store has ordered to replenish inventory and NOT the number of units that are currently on order by customers. Use this method when you only want to return the number of units on order for a specified item. If you want additional inventory information about an item, use the GetInventory (see page 1409) method. Below is example code that returns the number of units on order for an item with an ID of 12.

```
inventory.GetUnitsOnOrder(12);
```

C#

```
public virtual int GetUnitsOnOrder(long entryId);
```

Visual Basic

```
Public virtual Function GetUnitsOnOrder(entryId As long) As Integer
```

Parameters

Parameters	Description
long entryId	Enter the ID for a catalog entry (product).

Returns

Returns the number of units on order for a given catalog entry's ID.

Remarks

The default implementation uses GetInventory (see page 1409)().

InventoryProvider.IncreaseStockLevel Method

Increases the stock level for a catalog entry (product) based on its ID and the given quantity. For example, if you need to

increase the stock level of a given item (for example, item ID=12) by ten in your inventory system, you might use the following line of code:

```
inventory.IncreaseStockLevel(12, 10);
```

C#

```
public virtual void IncreaseStockLevel(long entryId, int quantity);
```

Visual Basic

```
Public virtual Function IncreaseStockLevel(entryId As long, quantity As Integer) As void
```

Parameters

Parameters	Description
long entryId	Enter the ID of the catalog entry (product).
int quantity	Enter the quantity by which the stock level will increase.

Remarks

The default implementation uses GetInventory (see page 1409)() and SaveInventory (see page 1411)().

InventoryProvider.SaveInventory Method

Saves inventory information for a catalog entry. Some of the items saved include:

- EntryId
- UnitsInStock
- UnitsOnOrder
- ReorderLevel

For example, if you have an item with the ID of 12 and you want to set its units in stock to 3, the units on order to 10, and the re-order level 5, you might use the following lines of code:

```
item.EntryId = 12;
item.UnitsInStock = 3;
item.UnitsOnOrder = 10;
item.ReorderLevel = 5;
inventory.SaveInventory(item);
```

When creating a custom inventory provider, you must implement this method.

C#

```
public abstract void SaveInventory(InventoryData inventory);
```

Visual Basic

```
Public abstract Function SaveInventory(inventory As InventoryData) As void
```

Parameters

Parameters	Description
inventoryData	The inventory object to save.

Remarks

Implementations of this method should raise the OnBeforeStockLevelSet, OnAfterStockLevelSet, and OnReOrderLevelReached events as appropriate. Other eCommerce (see page 601) functionality depends on these events occuring.

PaymentGatewayProvider Class

Base class for Payment Gateway Providers. A Payment Gateway Provider (see page 1407) is a pluggable component that is integrated into Ektron's CMS400.NET eCommerce (see page 601) module. A Payment provider handles eCommerce (see page 601) customer payments by utilizing third party payment gateways. CMS400.NET's eCommerce (see page 601) module accepts payments such as, credit cards. Then, it passes that information to a third party service. The third party service processes the payment, and returns a transaction ID that's stored with the customer's order.

You can use this base class to create your own custom payment gateway. For additional information on payment gateways, see the Ektron CMS400.NET Developer Manual's "Customizing eCommerce (see page 601)" > "Payment Gateway Providers" section.

Class Hierarchy

C#

```
public abstract class PaymentGatewayProvider : ProviderBase;
```

Visual Basic

```
Public abstract Class PaymentGatewayProvider
Inherits ProviderBase
```

File

PaymentProvider.cs

PaymentGatewayProvider Methods

The methods of the PaymentGatewayProvider class are listed here.

Public Methods

	Name	Description
	Authorize (see page 1412)	Authorizes a payment, but does not capture it. Returns an authorization code so it can be captured at a later date.
	AuthorizeAndCapture (see page 1413)	Authorizes and captures a payment.
	AuthorizeRecurringPayment (see page 1413)	Authorizes a recurring payment transaction.
	CancelRecurringPayment (see page 1413)	Cancels a recurring payment by providing its ID.
	CapturePreauthorization (see page 1414)	Captures a preauthorized payment.
	RecurringPaymentFailed (see page 1414)	Handles a recurring payment that has failed. This method should be the callback point when a recurring payment has failed.
	RecurringPaymentProcessed (see page 1414)	Handles a recurring payment that has been processed. This method should be the callback point when a recurring payment is processed.
	VoidPreAuthorization (see page 1414)	Voids a preauthorized payment.

PaymentGatewayProvider.Authorize Method

Authorizes a payment, but does not capture it. Returns an authorization code so it can be captured at a later date.

C#

```
public abstract string Authorize();
```

Visual Basic

```
Public abstract Function Authorize() As string
```

Returns

Returns the transaction ID of an authorization.

PaymentGatewayProvider.AuthorizeAndCapture Method

Authorizes and captures a payment.

C#

```
public abstract string AuthorizeAndCapture();
```

Visual Basic

```
Public abstract Function AuthorizeAndCapture() As string
```

Returns

Returns the transaction ID of an authorization.

PaymentGatewayProvider.AuthorizeRecurringPayment Method

Authorizes a recurring payment transaction.

C#

```
public virtual string AuthorizeRecurringPayment(RecurrenceData recurrence);
```

Visual Basic

```
Public virtual Function AuthorizeRecurringPayment(recurrence As RecurrenceData) As string
```

Parameters

Parameters	Description
RecurrenceData recurrence	Recurring payment schedule

Returns

Returns the recurring transaction ID.

PaymentGatewayProvider.CancelRecurringPayment Method

Cancels a recurring payment by providing its ID.

C#

```
public virtual void CancelRecurringPayment(string recurringTransactionId);
```

Visual Basic

```
Public virtual Function CancelRecurringPayment(recurringTransactionId As string) As void
```

Parameters

Parameters	Description
string recurringTransactionId	The ID of recurring transaction to cancel.

Returns

Returns the recurring transaction ID.

PaymentGatewayProvider.CapturePreauthorization Method

Captures a preauthorized payment.

C#

```
public abstract string CapturePreauthorization(string transactionId);
```

Visual Basic

```
Public abstract Function CapturePreauthorization(transactionId As string) As string
```

Returns

Returns the transaction ID of an authorization.

PaymentGatewayProvider.RecurringPaymentFailed Method

Handles a recurring payment that has failed. This method should be the callback point when a recurring payment has failed.

C#

```
public virtual void RecurringPaymentFailed(string recurringTransactionId);
```

Visual Basic

```
Public virtual Function RecurringPaymentFailed(recurringTransactionId As string) As void
```

Parameters

Parameters	Description
string recurringTransactionId	The ID of the recurring transaction.

PaymentGatewayProvider.RecurringPaymentProcessed Method

Handles a recurring payment that has been processed. This method should be the callback point when a recurring payment is processed.

C#

```
public virtual void RecurringPaymentProcessed(string recurringTransactionId, string
paymentTransactionId);
```

Visual Basic

```
Public virtual Function RecurringPaymentProcessed(recurringTransactionId As string,
paymentTransactionId As string) As void
```

Parameters

Parameters	Description
string recurringTransactionId	The ID of the recurring transaction.
string paymentTransactionId	The ID of the individual payment.

PaymentGatewayProvider.VoidPreAuthorization Method

Voids a preauthorized payment.

C#

```
public abstract string VoidPreAuthorization(string transactionId);
```

Visual Basic

```
Public abstract Function VoidPreAuthorization(transactionId As string) As string
```

Returns

Returns the transaction ID of an authorization.

PaymentGatewayProvider Properties

The properties of the PaymentGatewayProvider class are listed here.

Public Properties

	Name	Description
	Amount (see page 1415)	Gets or sets the payment amount being submitted.
	Authorization (see page 1416)	Gets or sets the PaymentAuthorization (see page 1534) result. Payment submission results should be set on the Authorization.
	Config (see page 1416)	Gets the PaymentProvider config result.
	CurrencyCode (see page 1416)	Gets or sets the currency's Alpha ISO Code.
	CurrencyId (see page 1416)	Gets or sets the currency's Numeric ISO Code.
	GatewayData (see page 1416)	Gets or sets the data associated with this payment gateway provider including the user ID and password for the gateway.
	InTestMode (see page 1416)	Gets or sets the "Test Mode" flag. True indicates the application is configured for testing.
	IsSubmissionSuccess (see page 1417)	Returns true if authorization was successful.
	OrderId (see page 1417)	Gets or sets the ID of the order associated with the payment submittal.
	PaymentMethod (see page 1417)	Gets or sets the Payment Method for this payment submittal. This could be a CreditCardPayment, eCheckPayment, etc.
	SubmissionError (see page 1417)	Returns any error messages that occurred during submission.
	SupportsCheckPayments (see page 1417)	Gets or sets whether the payment gateway supports check payments. • **True** = the payment gateway supports check payments • **False** = the payment gateway does not support check payments
	SupportsCreditCardPayments (see page 1418)	Gets or sets whether the payment gateway supports credit cards. • **True** = the payment gateway supports credit cards • **False** = the payment gateway does not support credit cards
	SupportsRecurringPayments (see page 1418)	Gets or sets the "Supports Recurring Payments" flag. If true, this provider supports recurring payments.

PaymentGatewayProvider.Amount Property

Gets or sets the payment amount being submitted.

C#

```
public virtual decimal Amount;
```

Visual Basic

```
Public virtual Property Amount() As decimal
```

PaymentGatewayProvider.Authorization Property

Gets or sets the PaymentAuthorization (see page 1534) result. Payment submission results should be set on the Authorization.

C#

```
public virtual PaymentAuthorization Authorization;
```

Visual Basic

```
Public virtual Property Authorization() As PaymentAuthorization
```

PaymentGatewayProvider.Config Property

Gets the PaymentProvider config result.

C#

```
public System.Collections.Specialized.NameValueCollection Config;
```

Visual Basic

```
Public Property Config() As System.Collections.Specialized.NameValueCollection
```

PaymentGatewayProvider.CurrencyCode Property

Gets or sets the currency's Alpha ISO Code.

C#

```
public virtual string CurrencyCode;
```

Visual Basic

```
Public virtual Property CurrencyCode() As string
```

PaymentGatewayProvider.CurrencyId Property

Gets or sets the currency's Numeric ISO Code.

C#

```
public virtual long CurrencyId;
```

Visual Basic

```
Public virtual Property CurrencyId() As long
```

PaymentGatewayProvider.GatewayData Property

Gets or sets the data associated with this payment gateway provider including the user ID and password for the gateway.

C#

```
public virtual PaymentGatewayData GatewayData;
```

Visual Basic

```
Public virtual Property GatewayData() As PaymentGatewayData
```

PaymentGatewayProvider.InTestMode Property

Gets or sets the "Test Mode" flag. True indicates the application is configured for testing.

C#

```
public virtual bool InTestMode;
```

Visual Basic

```
Public virtual Property InTestMode() As bool
```

PaymentGatewayProvider.IsSubmissionSuccess Property

Returns true if authorization was successful.

C#

```
public virtual bool IsSubmissionSuccess;
```

Visual Basic

```
Public virtual Property IsSubmissionSuccess() As bool
```

PaymentGatewayProvider.OrderId Property

Gets or sets the ID of the order associated with the payment submittal.

C#

```
public virtual long OrderId;
```

Visual Basic

```
Public virtual Property OrderId() As long
```

PaymentGatewayProvider.PaymentMethod Property

Gets or sets the Payment Method for this payment submittal. This could be a CreditCardPayment, eCheckPayment, etc.

C#

```
public virtual PaymentMethod PaymentMethod;
```

Visual Basic

```
Public virtual Property PaymentMethod() As PaymentMethod
```

PaymentGatewayProvider.SubmissionError Property

Returns any error messages that occurred during submission.

C#

```
public virtual string SubmissionError;
```

Visual Basic

```
Public virtual Property SubmissionError() As string
```

PaymentGatewayProvider.SupportsCheckPayments Property

Gets or sets whether the payment gateway supports check payments.

- **True** = the payment gateway supports check payments
- **False** = the payment gateway does not support check payments

C#

```
public virtual bool SupportsCheckPayments;
```

Visual Basic

```
Public virtual Property SupportsCheckPayments() As bool
```

PaymentGatewayProvider.SupportsCreditCardPayments Property

Gets or sets whether the payment gateway supports credit cards.

- **True** = the payment gateway supports credit cards
- **False** = the payment gateway does not support credit cards

C#

```
public virtual bool SupportsCreditCardPayments;
```

Visual Basic

```
Public virtual Property SupportsCreditCardPayments() As bool
```

PaymentGatewayProvider.SupportsRecurringPayments Property

Gets or sets the "Supports Recurring Payments" flag. If true, this provider supports recurring payments.

C#

```
public virtual bool SupportsRecurringPayments;
```

Visual Basic

```
Public virtual Property SupportsRecurringPayments() As bool
```

Ektron.Cms.Commerce.Shipment Namespace

This is namespace Ektron.Cms.Commerce.Shipment.

Namespaces

Name	Description
Provider (see page 1418)	This is namespace Ektron.Cms.Commerce.Shipment.Provider.

Ektron.Cms.Commerce.Shipment.Provider Namespace

This is namespace Ektron.Cms.Commerce.Shipment.Provider.

Classes

	Name	Description
	ShipmentProvider (see page 1419)	A Shipping Provider is a pluggable component integrated into Ektron's CMS400.NET eCommerce (see page 601) module. The Shipping provider handles eCommerce (see page 601) real-time shipping rate retrieval by utilizing third party shipping services such as FedEx or UPS, or you can create your own fixed rate logic. CMS400.NET's eCommerce (see page 601) shipping module calculates the package(s) needed for an order and then passes the following information to the company providing the shipping services via a Shipping Provider. • **package information** - for example, how many, height, width, depth and weight of each package. • **warehouse address** - the address from which the items will be shipped. A... more (see page 1419)

ShipmentProvider Class

A Shipping Provider (see page 1418) is a pluggable component integrated into Ektron's CMS400.NET eCommerce (see page 601) module. The Shipping provider handles eCommerce (see page 601) real-time shipping rate retrieval by utilizing third party shipping services such as FedEx or UPS, or you can create your own fixed rate logic. CMS400.NET's eCommerce (see page 601) shipping module calculates the package(s) needed for an order and then passes the following information to the company providing the shipping services via a Shipping Provider (see page 1418).

- **package information** - for example, how many, height, width, depth and weight of each package.
- **warehouse address** - the address from which the items will be shipped. A warehouse's information can be set in the Workarea > Modules > Commerce (see page 1423) > Shipping > Warehouses screen.
- **destination address** - the address a customer entered as his shipping address when making the purchase.
- **desired shipping options** - for example, if you have Next Day, 2-Day Ground and 3-Day Ground as shipping options, you can pass the option chosen by the customer.

CMS400.NET comes with several shipping providers, including FedEx and UPS. You can customize these providers or create your own using the extendable Shipping Provider (see page 1418) architecture.

Each type of shipping provider accepts configuration parameters. For example, FedEx requires a username, password, account number, and meter number while UPS requires a Username, password, and account number.

This base class allows you to define or retrieve information for a shipping provider.

Class Hierarchy

C#

```
public abstract class ShipmentProvider : ProviderBase;
```

Visual Basic

```
Public abstract Class ShipmentProvider
Inherits ProviderBase
```

File

ShipmentProvider.cs

ShipmentProvider Methods

The methods of the ShipmentProvider class are listed here.

Public Methods

	Name	Description
	GetRates (see page 1419)	Returns a list of shipping rates for the specified shipping methods.
	GetServiceTypes (see page 1420)	Returns a list of available service types supported by the shipment company.
	GetTrackingUrl (see page 1420)	Returns the Tracking URL for the provider.

ShipmentProvider.GetRates Method

Returns a list of shipping rates for the specified shipping methods.

C#

```
public abstract List<ShippingOptionData> GetRates(IEnumerable<ShippingMethodData>
expectedOptions, AddressData origin, AddressData destination, Weight weight, Dimensions
dimensions);
```

Visual Basic

```
Public abstract Function GetRates(expectedOptions As IEnumerable<ShippingMethodData>, origin
As AddressData, destination As AddressData, weight As Weight, dimensions As Dimensions) As
List<ShippingOptionData>
```

Parameters

Parameters	Description
AddressData origin	The address from which the package will be shipped.
AddressData destination	The address to which the package will be shipped.
Weight weight	The weight of the package being shipped.
Dimensions dimensions	The dimensions of the package being shipped.
shippingMethods	The shipping methods for which to retrieve rates.

Returns

Returns a list of shipping rates.

ShipmentProvider.GetServiceTypes Method

Returns a list of available service types supported by the shipment company.

C#

```
public abstract List<string> GetServiceTypes();
```

Visual Basic

```
Public abstract Function GetServiceTypes() As List<string>
```

Returns

Returns a list of available service types supported by the shipment company.

ShipmentProvider.GetTrackingUrl Method

Returns the Tracking URL for the provider.

C#

```
public abstract string GetTrackingUrl(string trackingId);
```

Visual Basic

```
Public abstract Function GetTrackingUrl(trackingId As string) As string
```

Parameters

Parameters	Description
string trackingId	The tracking identification number for your shipment which is provided by the shipping provider.

Returns

Returns the Tracking URL for the provider.

ShipmentProvider Properties

The properties of the ShipmentProvider class are listed here.

Public Properties

	Name	Description
	AccountNumber (see page 1421)	Gets or sets the account number associated with the account for the shipping company.
	IsTrackingSupported (see page 1421)	Gets or sets the "Tracking Supported" flag for this provider.

	Key (see page 1421)	Gets or sets the key associated with account for this provider. The key is typically a unique series of numbers and letters that the shipping company provides to you when sign up for an account.
	MeterNumber (see page 1421)	Gets or sets the meter number associated with the account for the shipping company. Not all shipping companies require a meter number. If one is not required, leave blank.
	Password (see page 1421)	Gets or sets the password associated with the account for the shipping company.
	ServiceUrl (see page 1422)	Gets or sets the URL used to contact the shipping company for rate requests. This URL is provided by the shipping company when sign up for an account.
	TransactionId (see page 1422)	Gets or sets the transaction ID associated with the account for the shipping company.

ShipmentProvider.AccountNumber Property

Gets or sets the account number associated with the account for the shipping company.

C#

```
public virtual string AccountNumber;
```

Visual Basic

```
Public virtual Property AccountNumber() As string
```

ShipmentProvider.IsTrackingSupported Property

Gets or sets the "Tracking Supported" flag for this provider.

C#

```
public virtual bool IsTrackingSupported;
```

Visual Basic

```
Public virtual Property IsTrackingSupported() As bool
```

ShipmentProvider.Key Property

Gets or sets the key associated with account for this provider. The key is typically a unique series of numbers and letters that the shipping company provides to you when sign up for an account.

C#

```
public virtual string Key;
```

Visual Basic

```
Public virtual Property Key() As string
```

ShipmentProvider.MeterNumber Property

Gets or sets the meter number associated with the account for the shipping company. Not all shipping companies require a meter number. If one is not required, leave blank.

C#

```
public virtual string MeterNumber;
```

Visual Basic

```
Public virtual Property MeterNumber() As string
```

ShipmentProvider.Password Property

Gets or sets the password associated with the account for the shipping company.

C#

```
public virtual string Password;
```

Visual Basic

```
Public virtual Property Password() As string
```

ShipmentProvider.ServiceUrl Property

Gets or sets the URL used to contact the shipping company for rate requests. This URL is provided by the shipping company when sign up for an account.

C#

```
public virtual string ServiceUrl;
```

Visual Basic

```
Public virtual Property ServiceUrl() As string
```

ShipmentProvider.TransactionId Property

Gets or sets the transaction ID associated with the account for the shipping company.

C#

```
public virtual string TransactionId;
```

Visual Basic

```
Public virtual Property TransactionId() As string
```

Data Classes

Classes

	Name	Description
	AssetUpdateData (see page 1606)	This is class Ektron.Cms.AssetUpdateData.
	ContentRatingLabels (see page 1611)	Provides a convenient data class for getting and setting Content (see page 822) Rating Labels.
	DiscussionCategory (see page 1614)	This is class Ektron.Cms.DiscussionCategory.
	FolderRequest (see page 1615)	This is class Ektron.Cms.FolderRequest.
	FormFieldDataItem (see page 1627)	Describes an instance of a data value from an individual form field. The data is a name/value pair.
	FormFieldDatalist (see page 1628)	Describes an individual datalist suitable as an enumeration for a select list.
	FormFieldDatalistItem (see page 1631)	Describes an individual datalist item (most importantly its value and display value).
	FormFieldDefinition (see page 1633)	Describes the definition of an individual form field.
	FormFieldList (see page 1636)	Describes the definition of all the fields in a specific form.
	FormRequestData (see page 1637)	A data object that contains information about a form.
	PagingInfo (see page 1641)	This is class Ektron.Cms.PagingInfo.
	TaxonomyRequest (see page 1643)	This is class Ektron.Cms.TaxonomyRequest.
	UriFavorite (see page 1646)	A list of properties that make up a URI favorite data.
	WorkSpaceGroupRequest (see page 1648)	This is class Ektron.Cms.WorkSpaceGroupRequest.
	WorkSpaceUserRequest (see page 1649)	This is class Ektron.Cms.WorkSpaceUserRequest.

Namespaces

Name	Description
Commerce (see page 1423)	This section includes data classes for eCommerce (see page 601). These classes get or set information about the different types of data contained in eCommerce (see page 601).
Common (see page 1589)	This is namespace Ektron.Cms.Common.

Ektron.Cms.Commerce Namespace

This section includes data classes for eCommerce (see page 601). These classes get or set information about the different types of data contained in eCommerce (see page 601).

Classes

	Name	Description
	AddressData (see page 1428)	Gets or sets information about a specified address. For example, if you were creating an address data object for a fictional user, Ivan Tory, at Ektron's address, the code might look like this:
	AdjustedBasketItem (see page 1431)	Class used to represent basket item price adjustments.
	AppliedCouponData (see page 1433)	This is class Ektron.Cms.Commerce.AppliedCouponData.

	AppliedCouponDataCollection (see page 1435)	This is class Ektron.Cms.Commerce.AppliedCouponDataCollection.
	AppliedCouponEventArgs (see page 1436)	This is class Ektron.Cms.Commerce.AppliedCouponEventArgs.
	Basket (see page 1437)	This is class Ektron.Cms.Commerce.Basket.
	BasketCalculatorData (see page 1442)	This is class Ektron.Cms.Commerce.BasketCalculatorData.
	BasketData (see page 1446)	This is class Ektron.Cms.Commerce.BasketData.
	BasketItem (see page 1449)	Data class representing a basket line item.
	BasketItemCollection (see page 1454)	Collection (see page 95) class for BasketItems.
	BasketVariantData (see page 1455)	
	BundleData (see page 1458)	
	CatalogData (see page 1459)	
	ContractData (see page 1460)	This is class Ektron.Cms.Commerce.ContractData.
	CountryData (see page 1461)	Gets or sets information about a specified country.
	CountryTaxRateData (see page 1464)	This is class Ektron.Cms.Commerce.CountryTaxRateData.
	CouponData (see page 1466)	This is class Ektron.Cms.Commerce.CouponData.
	CouponEntryData (see page 1470)	This is class Ektron.Cms.Commerce.CouponEntryData.
	CurrencyData (see page 1472)	Data class representing currencies in system.
	CurrencyPricingData (see page 1474)	Represents Pricing Information for agiven currency.
	CustomerData (see page 1478)	Data class representing Customers in system.
	Dimensions (see page 1482)	Class used to define dimension information.
	EntryAttributeData (see page 1485)	This is class Ektron.Cms.Commerce.EntryAttributeData.
	EntryData (see page 1487)	Base class for catalog entry data classes.
	EntryGroupData (see page 1499)	Data class representing Catalog Entry Groups in system.
	EntryGroupItemData (see page 1501)	This is class Ektron.Cms.Commerce.EntryGroupItemData.
	ExchangeRateData (see page 1503)	This is class Ektron.Cms.Commerce.ExchangeRateData.
	ImageMediaData (see page 1506)	This is class Ektron.Cms.ImageMediaData.
	InventoryData (see page 1506)	The InventoryData class allows you to get and set various inventory information for a catalog entry (product).
	KitConfigData (see page 1508)	Invitation class used for sending friend requests.
	KitData (see page 1510)	
	MediaData (see page 1510)	This is class Ektron.Cms.MediaData.
	MediaGalleryData (see page 1513)	This is class Ektron.Cms.MediaGalleryData.
	OptionGroupData (see page 1514)	Data class representing Option groups for Entry Kits.
	OptionGroupItemCollection (see page 1516)	Collection (see page 95) class for OptionGroupItemData (see page 1518).
	OptionGroupItemData (see page 1518)	Data class representing option group items in the CMS. Option Group Items are option items for a particular Kit Grouping.
	OrderData (see page 1520)	
	OrderLineData (see page 1526)	

	OrderPartData (see page 1528)	data class representing order part sin the cms.
	PackageData (see page 1532)	Data class used for representing package sizes used for shipping items.
	PaymentAuthorization (see page 1534)	Data class representing payment authorization from gateway.
	PaymentMethod (see page 1536)	Base class used for representing payment methods for use by payment provider.
	PipelineStageData (see page 1537)	
	PostalCodeTaxRateData (see page 1539)	This is class Ektron.Cms.Commerce.PostalCodeTaxRateData.
	PricingData (see page 1540)	This is class Ektron.Cms.Commerce.PricingData.
	ProductData (see page 1545)	
	ProductTypeAttributeData (see page 1546)	This is class Ektron.Cms.Commerce.ProductTypeAttributeData.
	ProductTypeData (see page 1549)	Data class representing ecommerce product types.
	RecommendationItemData (see page 1550)	Data class representing entry reccomendations in ecommerce system.
	RegionData (see page 1554)	Gets or sets information about specified geographic regions within countries that your eCommerce (see page 601) site services. For example, in the United States of America, each state is a region. So, if you were adding New Hampshire as a region, the code might look like this.
	RegionTaxRateData (see page 1557)	This is class Ektron.Cms.Commerce.RegionTaxRateData.
	ShippingBaseData (see page 1559)	Base data class for representing shipping method information.
	ShippingOptionData (see page 1560)	Data class representing a quote for a particular shipping method.
	TaxClassData (see page 1561)	Data object representing tax class information.
	TaxRateData (see page 1563)	This is class Ektron.Cms.Commerce.TaxRateData.
	TaxTypeData (see page 1565)	Data class representing Tax Types.
	ThumbnailData (see page 1568)	This is class Ektron.Cms.ThumbnailData.
	ThumbnailDefaultData (see page 1569)	This is class Ektron.Cms.ThumbnailDefaultData.
	TierPriceData (see page 1570)	This is class Ektron.Cms.Commerce.TierPriceData.
	ProductCouponEntryData (see page 1574)	This is class Ektron.Cms.Commerce.ProductCouponEntryData.
	WarehouseData (see page 1575)	This is class Ektron.Cms.Commerce.WarehouseData.
	Weight (see page 1577)	Class used to define mesurement information.
	ShippingMethodData (see page 1580)	Data class used to represent shipping methods defined in the cms.

Structs, Records, Enums

	Name	Description
	AddressProperty (see page 1582)	Enumeration of Address properties used for sorting and filtering AddressData (see page 1428) objects.
	CountryProperty (see page 1582)	Enumeration of CountryData (see page 1461) properties used for sorting and filtering CountryData (see page 1461) objects.
	CouponProperty (see page 1582)	Enumeration of Coupon properties used for sorting and filtering CouponData (see page 1466) objects.
	CurrencyProperty (see page 1583)	Enumeration of CurrencyData (see page 1472) properties used for sorting and filtering CurrencyData (see page 1472) objects.
	CustomerProperty (see page 1583)	Enumeration of Customer properties used for sorting and filtering CustomerData (see page 1478) objects.

	EntryAttributeProperty (see page 1583)	Enumeration of EntryAttributeProperty properties available for sorting and filtering.
	EntryGroupItemProperty (see page 1583)	Enumeration of Entry Group Item properties used for sorting and filtering EntryGroupItemData (see page 1501) objects.
	EntryGroupProperty (see page 1584)	Enumeration of EntryGroup properties used for sorting and filtering EntryGroupData (see page 1499) objects.
	EntryProperty (see page 1584)	Enumeration of EntryData (see page 1487) columns available for sorting and filtering.
	ExchangeRateProperty (see page 1584)	Enumeration of ExchangeRate columns available for sorting and filtering.
	LinearUnit (see page 1585)	Enumeration of units for measuring length.
	OrderProperty (see page 1585)	Enumeration of order properties used for sorting and filtering OrderData (see page 1520) objects.
	PackageProperty (see page 1585)	Enumeration representing properties of the packagedata class.
	ProductTypeAttributeProperty (see page 1586)	Enumeration of ProductTypeAttributeData (see page 1546) properties available for sorting and filtering.
	ProductTypeProperty (see page 1586)	Enumeration of ProductType properties used for sorting and filtering CustomerData (see page 1478) objects.
	RegionProperty (see page 1586)	Enumeration of Region properties used for sorting and filtering RegionData (see page 1554) objects.
	ShippingMethodProperty (see page 1586)	Enumeration representing properties of the ShippingMethodData (see page 1580) class.
	TaxClassProperty (see page 1587)	/ Enumeration of TaxClass columns available for sorting and filtering.
	TaxRateProperty (see page 1587)	Enumeration of possible values to order Lists of TaxRateData (see page 1563) by.
	TaxRateType (see page 1587)	This Represents the default Tax rate Types supported by The CMS. Additonal tax Types may be supported by customers.
	TaxTypeProperty (see page 1588)	/ Enumeration of TaxType columns available for sorting and filtering.
	WarehouseProperty (see page 1588)	Enumeration of Warehouse properties used for sorting and filtering WarehouseData (see page 1575) objects.
	WeightUnit (see page 1588)	Enumeration of units for measuring weight.

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	AddressData (see page 1428)	Gets or sets information about a specified address. For example, if you were creating an address data object for a fictional user, Ivan Tory, at Ektron's address, the code might look like this:
	AdjustedBasketItem (see page 1431)	Class used to represent basket item price adjustments.
	AppliedCouponData (see page 1433)	This is class Ektron.Cms.Commerce.AppliedCouponData.
	AppliedCouponDataCollection (see page 1435)	This is class Ektron.Cms.Commerce.AppliedCouponDataCollection.
	AppliedCouponEventArgs (see page 1436)	This is class Ektron.Cms.Commerce.AppliedCouponEventArgs.
	Basket (see page 1437)	This is class Ektron.Cms.Commerce.Basket.
	BasketCalculatorData (see page 1442)	This is class Ektron.Cms.Commerce.BasketCalculatorData.
	BasketData (see page 1446)	This is class Ektron.Cms.Commerce.BasketData.
	BasketItem (see page 1449)	Data class representing a basket line item.
	BasketItemCollection (see page 1454)	Collection (see page 95) class for BasketItems.

	BasketVariantData (see page 1455)	
	BundleData (see page 1458)	
	CatalogData (see page 1459)	
	ContractData (see page 1460)	This is class Ektron.Cms.Commerce.ContractData.
	CountryData (see page 1461)	Gets or sets information about a specified country.
	CountryTaxRateData (see page 1464)	This is class Ektron.Cms.Commerce.CountryTaxRateData.
	CouponData (see page 1466)	This is class Ektron.Cms.Commerce.CouponData.
	CouponEntryData (see page 1470)	This is class Ektron.Cms.Commerce.CouponEntryData.
	CurrencyData (see page 1472)	Data class representing currencies in system.
	CurrencyPricingData (see page 1474)	Represents Pricing Information for agiven currency.
	CustomerData (see page 1478)	Data class representing Customers in system.
	Dimensions (see page 1482)	Class used to define dimension information.
	EntryAttributeData (see page 1485)	This is class Ektron.Cms.Commerce.EntryAttributeData.
	EntryData (see page 1487)	Base class for catalog entry data classes.
	EntryGroupData (see page 1499)	Data class representing Catalog Entry Groups in system.
	EntryGroupItemData (see page 1501)	This is class Ektron.Cms.Commerce.EntryGroupItemData.
	ExchangeRateData (see page 1503)	This is class Ektron.Cms.Commerce.ExchangeRateData.
	ImageMediaData (see page 1506)	This is class Ektron.Cms.ImageMediaData.
	InventoryData (see page 1506)	The InventoryData class allows you to get and set various inventory information for a catalog entry (product).
	KitConfigData (see page 1508)	Invitation class used for sending friend requests.
	KitData (see page 1510)	
	MediaData (see page 1510)	This is class Ektron.Cms.MediaData.
	MediaGalleryData (see page 1513)	This is class Ektron.Cms.MediaGalleryData.
	OptionGroupData (see page 1514)	Data class representing Option groups for Entry Kits.
	OptionGroupItemCollection (see page 1516)	Collection (see page 95) class for OptionGroupItemData (see page 1518).
	OptionGroupItemData (see page 1518)	Data class representing option group items in the CMS. Option Group Items are option items for a particular Kit Grouping.
	OrderData (see page 1520)	
	OrderLineData (see page 1526)	
	OrderPartData (see page 1528)	data class representing order part sin the cms.
	PackageData (see page 1532)	Data class used for representing package sizes used for shipping items.
	PaymentAuthorization (see page 1534)	Data class representing payment authorization from gateway.
	PaymentMethod (see page 1536)	Base class used for representing payment methods for use by payment provider.
	PipelineStageData (see page 1537)	
	PostalCodeTaxRateData (see page 1539)	This is class Ektron.Cms.Commerce.PostalCodeTaxRateData.
	PricingData (see page 1540)	This is class Ektron.Cms.Commerce.PricingData.

	ProductData (see page 1545)	
	ProductTypeAttributeData (see page 1546)	This is class Ektron.Cms.Commerce.ProductTypeAttributeData.
	ProductTypeData (see page 1549)	Data class representing ecommerce product types.
	RecommendationItemData (see page 1550)	Data class representing entry reccomendations in ecommerce system.
	RegionData (see page 1554)	Gets or sets information about specified geographic regions within countries that your eCommerce (see page 601) site services. For example, in the United States of America, each state is a region. So, if you were adding New Hampshire as a region, the code might look like this.
	RegionTaxRateData (see page 1557)	This is class Ektron.Cms.Commerce.RegionTaxRateData.
	ShippingBaseData (see page 1559)	Base data class for representing shipping method information.
	ShippingOptionData (see page 1560)	Data class representing a quote for a particular shipping method.
	TaxClassData (see page 1561)	Data object representing tax class information.
	TaxRateData (see page 1563)	This is class Ektron.Cms.Commerce.TaxRateData.
	TaxTypeData (see page 1565)	Data class representing Tax Types.
	ThumbnailData (see page 1568)	This is class Ektron.Cms.ThumbnailData.
	ThumbnailDefaultData (see page 1569)	This is class Ektron.Cms.ThumbnailDefaultData.
	TierPriceData (see page 1570)	This is class Ektron.Cms.Commerce.TierPriceData.
	ProductCouponEntryData (see page 1574)	This is class Ektron.Cms.Commerce.ProductCouponEntryData.
	WarehouseData (see page 1575)	This is class Ektron.Cms.Commerce.WarehouseData.
	Weight (see page 1577)	Class used to define mesurement information.
	ShippingMethodData (see page 1580)	Data class used to represent shipping methods defined in the cms.

AddressData Class

Gets or sets information about a specified address. For example, if you were creating an address data object for a fictional user, Ivan Tory, at Ektron's address, the code might look like this:

```
Ektron.Cms.Commerce.AddressData address = new Ektron.Cms.Commerce.AddressData();
address.Name = "Ivan Tory";
address.Company = "Ektron, Inc.";
address.AddressLine1 = "542 Amherst Street (Route 101A)";
address.City = "Nashua";
address.Region = new Ektron.Cms.Commerce.RegionData();
address.Region.Id = 31;   // 31 is the region ID for New Hampshire.
address.PostalCode = "03063";
address.Country = new Ektron.Cms.Commerce.CountryData();
address.Country.Id = 840;   // 840 is the country ID for the United States
address.Phone = "603-594-0249";
AddressAPI.Add(address);
```

To update an existing address's information, use the `Id` (see page 1430) property to specify the address.

Class Hierarchy

C#

```
public class AddressData;
```

Visual Basic

```
Public Class AddressData
```

File

AddressData.cs

AddressData Properties

The properties of the AddressData class are listed here.

Public Properties

	Name	Description
	AddressLine1 (see page 1429)	Gets or sets the first line of the address. This property is required when creating an address data object.
	AddressLine2 (see page 1429)	Gets or sets line 2 of address.
	City (see page 1429)	Gets or sets the city associated with this address. This property is required when creating an address data object.
	Company (see page 1430)	Gets or sets the name of the company associated with this address. For example, if the address is for Ektron, enter "Ektron, Inc." in the property.
	Country (see page 1430)	Gets or sets the country associated with this address.
	Id (see page 1430)	Gets or sets the ID of an address.
	Name (see page 1430)	Gets or sets the name associated with the address. For example, if you are entering an address for Ivan Tory, enter the name "Ivan Tory" in this property. This property is required when creating an address data object.
	Phone (see page 1430)	Gets or sets the phone number associated with this address.
	PostalCode (see page 1430)	Gets or sets the postal code associated with this address. This property is required when creating an address data object.
	Region (see page 1431)	Gets or sets the region associated with this address.

AddressData.AddressLine1 Property

Gets or sets the first line of the address. This property is required when creating an address data object.

C#

```
public string AddressLine1;
```

Visual Basic

```
Public Property AddressLine1() As string
```

AddressData.AddressLine2 Property

Gets or sets line 2 of address.

C#

```
public string AddressLine2;
```

Visual Basic

```
Public Property AddressLine2() As string
```

AddressData.City Property

Gets or sets the city associated with this address. This property is required when creating an address data object.

C#

```
public string City;
```

Visual Basic

```
Public Property City() As string
```

AddressData.Company Property

Gets or sets the name of the company associated with this address. For example, if the address is for Ektron, enter "Ektron, Inc." in the property.

C#

```
public string Company;
```

Visual Basic

```
Public Property Company() As string
```

AddressData.Country Property

Gets or sets the country associated with this address.

C#

```
public CountryData Country;
```

Visual Basic

```
Public Property Country() As CountryData
```

AddressData.Id Property

Gets or sets the ID of an address.

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

AddressData.Name Property

Gets or sets the name associated with the address. For example, if you are entering an address for Ivan Tory, enter the name "Ivan Tory" in this property. This property is required when creating an address data object.

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

AddressData.Phone Property

Gets or sets the phone number associated with this address.

C#

```
public string Phone;
```

Visual Basic

```
Public Property Phone() As string
```

AddressData.PostalCode Property

Gets or sets the postal code associated with this address. This property is required when creating an address data object.

C#

```
public string PostalCode;
```

Visual Basic

```
Public Property PostalCode() As string
```

AddressData.Region Property

Gets or sets the region associated with this address.

C#

```
public RegionData Region;
```

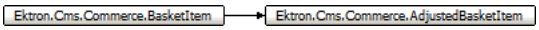
Visual Basic

```
Public Property Region() As RegionData
```

AdjustedBasketItem Class

Class used to represent basket item price adjustments.

Class Hierarchy



C#

```
public class AdjustedBasketItem : BasketItem;
```

Visual Basic

```
Public Class AdjustedBasketItem
Inherits BasketItem
```

File

AdjustedBasketItem.cs

AdjustedBasketItem Constructor

This is the overview for the AdjustedBasketItem constructor overload.

Overload List

	Name	Description
	AdjustedBasketItem.AdjustedBasketItem () (see page 1431)	This is AdjustedBasketItem, a member of class AdjustedBasketItem.
	AdjustedBasketItem.AdjustedBasketItem (BasketItem) (see page 1432)	Creates a new AdjustedBasketItem based upon base basketitem.

AdjustedBasketItem.AdjustedBasketItem Constructor ()

C#

```
public AdjustedBasketItem();
```

Visual Basic

```
Public Sub New()
```

Description

This is AdjustedBasketItem, a member of class AdjustedBasketItem.

AdjustedBasketItem.AdjustedBasketItem Constructor (BasketItem)

Creates a new AdjustedBasketItem based upon base basketitem.

C#

```
public AdjustedBasketItem(BasketItem basketItem);
```

Visual Basic

```
Public Sub New(basketItem As BasketItem)
```

Parameters

Parameters	Description
BasketItem basketItem	

AdjustedBasketItem Properties

The properties of the AdjustedBasketItem class are listed here.

Public Properties

	Name	Description
	AdjustedTotal (see page 1432)	Gets or sets the Total Adjusted price for the line item. This could reflect any coupon discounts, surcharges, etc applied to the item.
	ApplicableTaxRate (see page 1432)	Gets or sets the TaxRate class applicable to this item.
	AppliedCoupons (see page 1433)	Gets the list coupons applied to this basket item.
	HasFreeShipping (see page 1433)	Gets or sets whether or not this item has a free shipping coupon applied.
	TaxesApplied (see page 1433)	Gets or sets the amount of tax applicable to this item.
	TotalCouponDiscount (see page 1433)	Gets or sets the total coupon discount for this basket item.

AdjustedBasketItem.AdjustedTotal Property

Gets or sets the Total Adjusted price for the line item. This could reflect any coupon discounts, surcharges, etc applied to the item.

C#

```
public override decimal AdjustedTotal;
```

Visual Basic

```
Public override Property AdjustedTotal() As decimal
```

AdjustedBasketItem.ApplicableTaxRate Property

Gets or sets the TaxRate class applicable to this item.

C#

```
public TaxRateData ApplicableTaxRate;
```

Visual Basic

```
Public Property ApplicableTaxRate() As TaxRateData
```

AdjustedBasketItem.AppliedCoupons Property

Gets the list coupons applied to this basket item.

C#

```
public AppliedCouponDataCollection AppliedCoupons;
```

Visual Basic

```
Public Property AppliedCoupons() As AppliedCouponDataCollection
```

AdjustedBasketItem.HasFreeShipping Property

Gets or sets whether or not this item has a free shipping coupon applied.

C#

```
public bool HasFreeShipping;
```

Visual Basic

```
Public Property HasFreeShipping() As bool
```

AdjustedBasketItem.TaxesApplied Property

Gets or sets the amount of tax applicable to this item.

C#

```
public decimal TaxesApplied;
```

Visual Basic

```
Public Property TaxesApplied() As decimal
```

AdjustedBasketItem.TotalCouponDiscount Property

Gets or sets the total coupon discount for this basket item.

C#

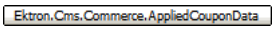
```
public decimal TotalCouponDiscount;
```

Visual Basic

```
Public Property TotalCouponDiscount() As decimal
```

AppliedCouponData Class

Class Hierarchy



C#

```
public class AppliedCouponData;
```

Visual Basic

```
Public Class AppliedCouponData
```

File

BasketItemCouponData.cs

Description

This is class Ektron.Cms.Commerce.AppliedCouponData.

AppliedCouponData Constructor

This is the overview for the AppliedCouponData constructor overload.

Overload List

	Name	Description
	AppliedCouponData.AppliedCouponData () (see page 1434)	This is AppliedCouponData, a member of class AppliedCouponData.
	AppliedCouponData.AppliedCouponData (CouponData, decimal) (see page 1434)	

AppliedCouponData.AppliedCouponData Constructor ()

C#

```
public AppliedCouponData();
```

Visual Basic

```
Public Sub New()
```

Description

This is AppliedCouponData, a member of class AppliedCouponData.

AppliedCouponData.AppliedCouponData Constructor (CouponData, decimal)

C#

```
public AppliedCouponData(CouponData coupon, decimal discount);
```

Visual Basic

```
Public Sub New(coupon As CouponData, discount As decimal)
```

Parameters

Parameters	Description
CouponData coupon	sets the coupon being applied to the basketItem.
decimal discount	sets the amount the coupon deducts from the price.

AppliedCouponData Properties

The properties of the AppliedCouponData class are listed here.

Public Properties

	Name	Description
	Coupon (see page 1434)	Gets or sets the coupon being applied to the basketItem.
	CouponDiscount (see page 1435)	Gets or sets the amount the coupon deducts from the price.

AppliedCouponData.Coupon Property

Gets or sets the coupon being applied to the basketItem.

C#

```
public CouponData Coupon;
```

Visual Basic

```
Public Property Coupon() As CouponData
```

AppliedCouponData.CouponDiscount Property

Gets or sets the amount the coupon deducts from the price.

C#

```
public decimal CouponDiscount;
```

Visual Basic

```
Public Property CouponDiscount() As decimal
```

AppliedCouponDataCollection Class

Class Hierarchy

C#

```
public class AppliedCouponDataCollection : Collection<AppliedCouponData>;
```

Visual Basic

```
Public Class AppliedCouponDataCollection
Inherits Collection
```

File

BasketItemCouponData.cs

Description

This is class Ektron.Cms.Commerce.AppliedCouponDataCollection.

AppliedCouponDataCollection Fields

The fields of the AppliedCouponDataCollection class are listed here.

Public Fields

	Name	Description
	ItemAdded (see page 1435)	This is ItemAdded, a member of class AppliedCouponDataCollection.

AppliedCouponDataCollection.ItemAdded Field

C#

```
public EventHandler<AppliedCouponEventArgs> ItemAdded;
```

Visual Basic

```
Public ItemAdded As EventHandler<AppliedCouponEventArgs>
```

Description

This is ItemAdded, a member of class AppliedCouponDataCollection.

AppliedCouponDataCollection Methods

The methods of the AppliedCouponDataCollection class are listed here.

Public Methods

	Name	Description
	FindByCouponId (see page 1436)	This is FindByCouponId, a member of class AppliedCouponDataCollection.

AppliedCouponDataCollection.FindByCouponId Method

C#

```
public AppliedCouponData FindByCouponId(long id);
```

Visual Basic

```
Public Function FindByCouponId(id As long) As AppliedCouponData
```

Description

This is FindByCouponId, a member of class AppliedCouponDataCollection.

AppliedCouponEventArgs Class

Class Hierarchy

C#

```
public class AppliedCouponEventArgs : EventArgs;
```

Visual Basic

```
Public Class AppliedCouponEventArgs
Inherits EventArgs
```

File

BasketItemCouponData.cs

Description

This is class Ektron.Cms.Commerce.AppliedCouponEventArgs.

AppliedCouponEventArgs Constructor

This is the overview for the AppliedCouponEventArgs constructor overload.

Overload List

	Name	Description
	AppliedCouponEventArgs.AppliedCouponEventArgs () (see page 1436)	This is AppliedCouponEventArgs, a member of class AppliedCouponEventArgs.
	AppliedCouponEventArgs.AppliedCouponEventArgs (AppliedCouponData) (see page 1437)	This is AppliedCouponEventArgs, a member of class AppliedCouponEventArgs.

AppliedCouponEventArgs.AppliedCouponEventArgs Constructor ()

C#

```
public AppliedCouponEventArgs();
```

Visual Basic

```
Public Sub New()
```

Description

This is AppliedCouponEventArgs, a member of class AppliedCouponEventArgs.

AppliedCouponEventArgs.AppliedCouponEventArgs Constructor (AppliedCouponData)

C#

```
public AppliedCouponEventArgs(AppliedCouponData appliedCoupon);
```

Visual Basic

```
Public Sub New(appliedCoupon As AppliedCouponData)
```

Description

This is AppliedCouponEventArgs, a member of class AppliedCouponEventArgs.

AppliedCouponEventArgs Properties

The properties of the AppliedCouponEventArgs class are listed here.

Public Properties

	Name	Description
	AppliedCoupon (see page 1437)	This is AppliedCoupon, a member of class AppliedCouponEventArgs.

AppliedCouponEventArgs.AppliedCoupon Property

C#

```
public AppliedCouponData AppliedCoupon;
```

Visual Basic

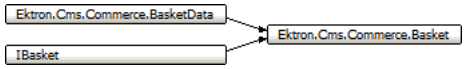
```
Public Property AppliedCoupon() As AppliedCouponData
```

Description

This is AppliedCoupon, a member of class AppliedCouponEventArgs.

Basket Class

Class Hierarchy



C#

```
public abstract class Basket : BasketData, IBasket;
```

Visual Basic

```
Public abstract Class Basket
Inherits BasketData
Inherits IBasket
```

File

IBasket.cs

Description

This is class Ektron.Cms.Commerce.Basket.

Basket Methods

The methods of the Basket class are listed here.

Public Methods

	Name	Description
	AddProduct (see page 1438)	This is the overview for the AddProduct method overload.
	AddProductVariant (see page 1439)	This is AddProductVariant, a member of class Basket.
	Decrement (see page 1439)	This is Decrement, a member of class Basket.
	GetItemIndex (see page 1440)	This is GetItemIndex, a member of class Basket.
	GetKitConfiguration (see page 1440)	This is GetKitConfiguration, a member of class Basket.
	GetKitConfigurationAsAdmin (see page 1440)	This is GetKitConfigurationAsAdmin, a member of class Basket.
	IncrementItemQuantity (see page 1440)	This is IncrementItemQuantity, a member of class Basket.
	MakeDefaultBasket (see page 1440)	This is MakeDefaultBasket, a member of class Basket.
	RemoveItem (see page 1441)	This is RemoveItem, a member of class Basket.
	SaveBasket (see page 1441)	This is SaveBasket, a member of class Basket.
	UpdateItemQuantity (see page 1441)	This is UpdateItemQuantity, a member of class Basket.
	UpdateKitConfiguration (see page 1441)	This is UpdateKitConfiguration, a member of class Basket.

AddProduct Method

This is the overview for the AddProduct method overload.

Overload List

	Name	Description
	Basket.AddProduct (long) (see page 1438)	This is AddProduct, a member of class Basket.
	Basket.AddProduct (long, int) (see page 1439)	Adds product to cart.
	Basket.AddProduct (long, KitConfigData) (see page 1439)	This is AddProduct, a member of class Basket.

Basket.AddProduct Method (long)

C#

```
public abstract long AddProduct(long productId);
```

Visual Basic

```
Public abstract Function AddProduct(productId As long) As long
```

Description

This is AddProduct, a member of class Basket.

Basket.AddProduct Method (long, KitConfigData)

C#

```
public abstract long AddProduct(long productId, KitConfigData config);
```

Visual Basic

```
Public abstract Function AddProduct(productId As long, config As KitConfigData) As long
```

Description

This is AddProduct, a member of class Basket.

Basket.AddProduct Method (long, int)

Adds product to cart.

C#

```
public abstract long AddProduct(long productId, int quantity);
```

Visual Basic

```
Public abstract Function AddProduct(productId As long, quantity As Integer) As long
```

Parameters

Parameters	Description
long productId	Id (see page 1447) of product to add to basket.
int quantity	quantity of product to add to cart.

Returns

returns basket item Id (see page 1447).

Basket.AddProductVariant Method

C#

```
public abstract long AddProductVariant(long productId, long variantId);
```

Visual Basic

```
Public abstract Function AddProductVariant(productId As long, variantId As long) As long
```

Description

This is AddProductVariant, a member of class Basket.

Basket.Decrement Method

C#

```
public abstract void Decrement(long basketItemId);
```

Visual Basic

```
Public abstract Function Decrement(basketItemId As long) As void
```

Description

This is Decrement, a member of class Basket.

Basket.GetItemIndex Method

C#

```
public abstract int GetItemIndex(long basketItemId);
```

Visual Basic

```
Public abstract Function GetItemIndex(basketItemId As long) As Integer
```

Description

This is GetItemIndex, a member of class Basket.

Basket.GetKitConfiguration Method

C#

```
public abstract KitConfigData GetKitConfiguration(long basketItemId);
```

Visual Basic

```
Public abstract Function GetKitConfiguration(basketItemId As long) As KitConfigData
```

Description

This is GetKitConfiguration, a member of class Basket.

Basket.GetKitConfigurationAsAdmin Method

C#

```
public abstract KitConfigData GetKitConfigurationAsAdmin(long basketItemId);
```

Visual Basic

```
Public abstract Function GetKitConfigurationAsAdmin(basketItemId As long) As KitConfigData
```

Description

This is GetKitConfigurationAsAdmin, a member of class Basket.

Basket.IncrementItemQuantity Method

C#

```
public abstract void IncrementItemQuantity(long basketItemId);
```

Visual Basic

```
Public abstract Function IncrementItemQuantity(basketItemId As long) As void
```

Description

This is IncrementItemQuantity, a member of class Basket.

Basket.MakeDefaultBasket Method

C#

```
public abstract void MakeDefaultBasket();
```

Visual Basic

```
Public abstract Function MakeDefaultBasket() As void
```

Description

This is MakeDefaultBasket, a member of class Basket.

Basket.RemoveItem Method

C#

```
public abstract void RemoveItem(long basketItemId);
```

Visual Basic

```
Public abstract Function RemoveItem(basketItemId As long) As void
```

Description

This is RemoveItem, a member of class Basket.

Basket.SaveBasket Method

C#

```
public abstract void SaveBasket();
```

Visual Basic

```
Public abstract Function SaveBasket() As void
```

Description

This is SaveBasket, a member of class Basket.

Basket.UpdateItemQuantity Method

C#

```
public abstract void UpdateItemQuantity(long basketItemId, int productQuantity);
```

Visual Basic

```
Public abstract Function UpdateItemQuantity(basketItemId As long, productQuantity As Integer)
As void
```

Description

This is UpdateItemQuantity, a member of class Basket.

Basket.UpdateKitConfiguration Method

C#

```
public abstract long UpdateKitConfiguration(long productId, KitConfigData config);
```

Visual Basic

```
Public abstract Function UpdateKitConfiguration(productId As long, config As KitConfigData) As
long
```

Description

This is UpdateKitConfiguration, a member of class Basket.

Basket Properties

The properties of the Basket class are listed here.

Public Properties

	Name	Description
	RequestInformation (see page 1442)	This is RequestInformation, a member of class Basket.

Basket.RequestInformation Property

C#

```
public abstract EkRequestInformation RequestInformation;
```

Visual Basic

```
Public abstract Property RequestInformation() As EkRequestInformation
```

Description

This is RequestInformation, a member of class Basket.

BasketCalculatorData Class

Class Hierarchy

Ektron.Cms.Commerce.BasketCalculatorData

C#

```
public class BasketCalculatorData;
```

Visual Basic

```
Public Class BasketCalculatorData
```

File

BasketCalculatorData.cs

Description

This is class Ektron.Cms.Commerce.BasketCalculatorData.

BasketCalculatorData Constructor

This is the overview for the BasketCalculatorData constructor overload.

Overload List

	Name	Description
	BasketCalculatorData.BasketCalculatorData () (see page 1443)	This is BasketCalculatorData, a member of class BasketCalculatorData.
	BasketCalculatorData.BasketCalculatorData (BasketData) (see page 1443)	This is BasketCalculatorData, a member of class BasketCalculatorData.

BasketCalculatorData.BasketCalculatorData Constructor ()

C#

```
public BasketCalculatorData();
```

Visual Basic

```
Public Sub New()
```

Description

This is BasketCalculatorData, a member of class BasketCalculatorData.

BasketCalculatorData.BasketCalculatorData Constructor (BasketData)

C#

```
public BasketCalculatorData(BasketData basket);
```

Visual Basic

```
Public Sub New(basket As BasketData)
```

Description

This is BasketCalculatorData, a member of class BasketCalculatorData.

BasketCalculatorData Properties

The properties of the BasketCalculatorData class are listed here.

Public Properties

	Name	Description
	AdjustedItems (see page 1444)	Gets the list of adjusted basket items in the current basket.
	AllBasketCoupons (see page 1444)	Gets or sets AllBasketCoupons. These are all coupons that have been applied to the basket regardless of the effect they have on the basket subtotal.
	Basket (see page 1444)	Gets or sets the Basket being calculated by the BasketCalculator.
	BasketCoupons (see page 1444)	Gets or sets BasketCoupons. These are all coupons that have been applied to the basket and their cumulative discount. If a coupon applies to more than one item, the discount amount for the coupon would be the total amount.
	BasketTotal (see page 1444)	Gets the basket total based upon item prices, coupons, taxes, shipping and any other adjustments.
	HasFreeShipping (see page 1444)	Gets or sets whether or not this basket has afree shipping coupon applied.
	ShipFrom (see page 1445)	Gets or sets the address from which the order would ship. This could be used for shipping, tax calculations, etc. If it's ommited, the shipping and tax information will not be calculated.
	ShippingRates (see page 1445)	Gets a list of shipping rates for the basket.
	ShipTo (see page 1445)	Gets or sets the address that the order is shipping to. This could be used for shipping, tax calculations, etc. If it's ommited, the shipping and tax information will not be calculated.
	TaxClassSubtotals (see page 1445)	Gets a dictionary representing the tax classes applicable to the basket with the amount of taxes applied to the basket for that tax class.
	TotalCouponDiscount (see page 1445)	Gets the total amount of coupons calculated for the current basket
	TotalShippingCost (see page 1446)	Gets the total amount of shipping costs calculated for the current basket.
	TotalTaxes (see page 1446)	Gets the total amount of taxes calculated for the current basket.
	ValidationResults (see page 1446)	Gets any ValidationResults encountered during calculation.

BasketCalculatorData.AdjustedItems Property

Gets the list of adjusted basket items in the current basket.

C#

```
public List<AdjustedBasketItem> AdjustedItems;
```

Visual Basic

```
Public Property AdjustedItems() As List<AdjustedBasketItem>
```

BasketCalculatorData.AllBasketCoupons Property

Gets or sets AllBasketCoupons. These are all coupons that have been applied to the basket regardless of the effect they have on the basket subtotal.

C#

```
public List<CouponData> AllBasketCoupons;
```

Visual Basic

```
Public Property AllBasketCoupons() As List<CouponData>
```

BasketCalculatorData.Basket Property

Gets or sets the Basket being calculated by the BasketCalculator.

C#

```
public BasketData Basket;
```

Visual Basic

```
Public Property Basket() As BasketData
```

BasketCalculatorData.BasketCoupons Property

Gets or sets BasketCoupons. These are all coupons that have been applied to the basket and their cumulative discount. If a coupon applies to more than one item, the discount amount for the coupon would be the total amount.

C#

```
public AppliedCouponDataCollection BasketCoupons;
```

Visual Basic

```
Public Property BasketCoupons() As AppliedCouponDataCollection
```

BasketCalculatorData.BasketTotal Property

Gets the basket total based upon item prices, coupons, taxes, shipping and any other adjustments.

C#

```
public decimal BasketTotal;
```

Visual Basic

```
Public Property BasketTotal() As decimal
```

BasketCalculatorData.HasFreeShipping Property

Gets or sets whether or not this basket has afree shipping coupon applied.

C#

```
public bool HasFreeShipping;
```

Visual Basic

```
Public Property HasFreeShipping() As bool
```

BasketCalculatorData.ShipFrom Property

Gets or sets the address from which the order would ship. This could be used for shipping, tax calculations, etc. If it's ommited, the shipping and tax information will not be calculated.

C#

```
public AddressData ShipFrom;
```

Visual Basic

```
Public Property ShipFrom() As AddressData
```

BasketCalculatorData.ShippingRates Property

Gets a list of shipping rates for the basket.

C#

```
public List<ShippingOptionData> ShippingRates;
```

Visual Basic

```
Public Property ShippingRates() As List<ShippingOptionData>
```

BasketCalculatorData.ShipTo Property

Gets or sets the address that the order is shipping to. This could be used for shipping, tax calculations, etc. If it's ommited, the shipping and tax information will not be calculated.

C#

```
public AddressData ShipTo;
```

Visual Basic

```
Public Property ShipTo() As AddressData
```

BasketCalculatorData.TaxClassSubtotals Property

Gets a dictionary representing the tax classes applicable to the basket with the amount of taxes applied to the basket for that tax class.

C#

```
public Dictionary<TaxClassData, decimal> TaxClassSubtotals;
```

Visual Basic

```
Public Property TaxClassSubtotals() As Dictionary<TaxClassData, decimal>
```

BasketCalculatorData.TotalCouponDiscount Property

Gets the total amount of coupons calculated for the current basket

C#

```
public decimal TotalCouponDiscount;
```

Visual Basic

```
Public Property TotalCouponDiscount() As decimal
```

BasketCalculatorData.TotalShippingCost Property

Gets the total amount of shipping costs calculated for the current basket.

C#

```
public decimal TotalShippingCost;
```

Visual Basic

```
Public Property TotalShippingCost() As decimal
```

BasketCalculatorData.TotalTaxes Property

Gets the total amount of taxes calculated for the current basket.

C#

```
public decimal TotalTaxes;
```

Visual Basic

```
Public Property TotalTaxes() As decimal
```

BasketCalculatorData.ValidationResults Property

Gets any ValidationResults encountered during calculation.

C#

```
public ValidationResults ValidationResults;
```

Visual Basic

```
Public Property ValidationResults() As ValidationResults
```

BasketData Class

Class Hierarchy

Ektron.Cms.Commerce.BasketData

C#

```
public class BasketData;
```

Visual Basic

```
Public Class BasketData
```

File

BasketData.cs

Description

This is class Ektron.Cms.Commerce.BasketData.

BasketData Properties

The properties of the BasketData class are listed here.

Public Properties

	Name	Description
	DateCreated (see page 1447)	Gets or sets the date created.

	DateModified (see page 1447)	Gets or sets the date modified.
	Id (see page 1447)	Gets or sets the Id of the Basket (see page 1437).
	IsDefault (see page 1447)	Gets or sets value indicating whether or not this basket is the default basket for customers.
	Items (see page 1447)	Gets the collection items in the basket.
	Name (see page 1448)	Gets or sets the name of the basket.
	ShippingAddressId (see page 1448)	Gets or sets the Id (see page 1447) of the address this basket will be shipping to. This is used for calculating taxes, shipping, etc.
	ShippingMethodId (see page 1448)	Gets or sets the id of the shipping method to be used for the basket. This is used for calculating shipping costs.
	ShopperId (see page 1448)	Gets or sets the Shoppervisitor Id (see page 1447) using the basket. This Id (see page 1447) is used when customer is not logged in.
	Subtotal (see page 1448)	Gets the basket subtotal.
	UserId (see page 1449)	Gets or sets the Id (see page 1447) of the user asscoiated with the basket.

BasketData.DateCreated Property

Gets or sets the date created.

C#

```
public DateTime DateCreated;
```

Visual Basic

```
Public Property DateCreated() As DateTime
```

BasketData.DateModified Property

Gets or sets the date modified.

C#

```
public DateTime DateModified;
```

Visual Basic

```
Public Property DateModified() As DateTime
```

BasketData.Id Property

Gets or sets the Id of the Basket (see page 1437).

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

BasketData.IsDefault Property

Gets or sets value indicating whether or not this basket is the default basket for customers.

C#

```
public bool IsDefault;
```

Visual Basic

```
Public Property IsDefault() As bool
```

BasketData.Items Property

Gets the collection items in the basket.

C#

```
public BasketItemCollection Items;
```

Visual Basic

```
Public Property Items() As BasketItemCollection
```

BasketData.Name Property

Gets or sets the name of the basket.

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

BasketData.ShippingAddressId Property

Gets or sets the Id (see page 1447) of the address this basket will be shipping to. This is used for calculating taxes, shipping, etc.

C#

```
public long ShippingAddressId;
```

Visual Basic

```
Public Property ShippingAddressId() As long
```

BasketData.ShippingMethodId Property

Gets or sets the id of the shipping method to be used for the basket. This is used for calculating shipping costs.

C#

```
public long ShippingMethodId;
```

Visual Basic

```
Public Property ShippingMethodId() As long
```

BasketData.ShopperId Property

Gets or sets the Shoppervisitor Id (see page 1447) using the basket. This Id (see page 1447) is used when customer is not logged in.

C#

```
public string ShopperId;
```

Visual Basic

```
Public Property ShopperId() As string
```

BasketData.Subtotal Property

Gets the basket subtotal.

C#

```
public decimal Subtotal;
```

Visual Basic

```
Public Property Subtotal() As decimal
```

BasketData.UserId Property

Gets or sets the Id (see page 1447) of the user asscoiated with the basket.

C#

```
public long UserId;
```

Visual Basic

```
Public Property UserId() As long
```

BasketItem Class

Data class representing a basket line item.

Class Hierarchy

Ektron.Cms.Commerce.BasketItem

C#

```
public class BasketItem;
```

Visual Basic

```
Public Class BasketItem
```

File

BasketItem.cs

BasketItem Constructor

This is the overview for the BasketItem constructor overload.

Overload List

	Name	Description
	BasketItem.BasketItem () (see page 1449)	This is BasketItem, a member of class BasketItem.
	BasketItem.BasketItem (BasketItem) (see page 1449)	constructor used for cloning Existing BasketItem.

BasketItem.BasketItem Constructor ()

C#

```
public BasketItem();
```

Visual Basic

```
Public Sub New()
```

Description

This is BasketItem, a member of class BasketItem.

BasketItem.BasketItem Constructor (BasketItem)

constructor used for cloning Existing BasketItem.

C#

```
public BasketItem(BasketItem basketItem);
```

Visual Basic

```
Public Sub New(basketItem As BasketItem)
```

Parameters

Parameters	Description
BasketItem basketItem	

BasketItem Properties

The properties of the BasketItem class are listed here.

Public Properties

	Name	Description
	AdjustedTotal (see page 1450)	Gets or sets the Total Adjusted price for the line item.
	Configuration (see page 1450)	Gets or sets any applicable Kit configuarations for the product in the basket for this line item.
	Dimensions (see page 1451)	Gets or sets the dimensions of the product. Used for shipping.
	Id (see page 1451)	Gets or sets the Id of this line item.
	IsInventoryManagementDisabled (see page 1451)	Gets or sets the disable inventory management flag. If true, this cart item does not track inventory and no inventory management is performed.
	IsTangible (see page 1451)	Indicates if this product is a tangible good.
	ListPrice (see page 1451)	Gets or sets the list price for the line item.
	Pricing (see page 1452)	The item pricing, including currencies and quantity.
	ProductId (see page 1452)	Gets or sets the Product (see page 431) Id (see page 1451) associated with this this line item.
	ProductLanguage (see page 1452)	Gets or sets the language for the product in the basket for this line item.
	ProductSku (see page 1452)	Gets or sets the Sku for the product in the basket for this line item.
	ProductTitle (see page 1452)	Gets or sets the Product (see page 431) Title for the product in the basket for this line item.
	ProductType (see page 1453)	Gets or sets the type of the product in the basket for this line item.
	Quantity (see page 1453)	Gets or sets the quantity of the product in the basket for this line item.
	Quicklink (see page 1453)	Gets or sets the product's Quicklink.
	SalePrice (see page 1453)	Sets the base sale price for the line item. The returned sale price will reflect any adjustments based upon kit options or product variants selected.
	TaxClassId (see page 1453)	Gets or sets the tax class Id (see page 1451) for the product in the basket for this line item.
	Variant (see page 1453)	Gets or sets and applicable varioant information for the product in the basket for this line item.
	Weight (see page 1454)	Gets or sets the weight of the product. Used for shipping.

BasketItem.AdjustedTotal Property

Gets or sets the Total Adjusted price for the line item.

C#

```
public virtual decimal AdjustedTotal;
```

Visual Basic

```
Public virtual Property AdjustedTotal() As decimal
```

BasketItem.Configuration Property

Gets or sets any applicable Kit configuarations for the product in the basket for this line item.

C#

```
public KitConfigData Configuration;
```

Visual Basic

```
Public Property Configuration() As KitConfigData
```

BasketItem.Dimensions Property

Gets or sets the dimensions of the product. Used for shipping.

C#

```
public Dimensions Dimensions;
```

Visual Basic

```
Public Property Dimensions() As Dimensions
```

BasketItem.Id Property

Gets or sets the Id of this line item.

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

BasketItem.IsInventoryManagementDisabled Property

Gets or sets the disable inventory management flag. If true, this cart item does not track inventory and no inventory management is performed.

C#

```
public bool IsInventoryManagementDisabled;
```

Visual Basic

```
Public Property IsInventoryManagementDisabled() As bool
```

BasketItem.IsTangible Property

Indicates if this product is a tangible good.

C#

```
public bool IsTangible;
```

Visual Basic

```
Public Property IsTangible() As bool
```

Returns

Boolean value indicating if this product is a tangible good.

BasketItem.ListPrice Property

Gets or sets the list price for the line item.

C#

```
public decimal ListPrice;
```

Visual Basic

```
Public Property ListPrice() As decimal
```

BasketItem.Pricing Property

The item pricing, including currencies and quantity.

C#

```
public PricingData Pricing;
```

Visual Basic

```
Public Property Pricing() As PricingData
```

Returns

PricingData (see page 1540) of the basket item.

BasketItem.ProductId Property

Gets or sets the Product (see page 431) Id (see page 1451) associated with this this line item.

C#

```
public long ProductId;
```

Visual Basic

```
Public Property ProductId() As long
```

BasketItem.ProductLanguage Property

Gets or sets the language for the product in the basket for this line item.

C#

```
public long ProductLanguage;
```

Visual Basic

```
Public Property ProductLanguage() As long
```

BasketItem.ProductSku Property

Gets or sets the Sku for the product in the basket for this line item.

C#

```
public string ProductSku;
```

Visual Basic

```
Public Property ProductSku() As string
```

BasketItem.ProductTitle Property

Gets or sets the Product (see page 431) Title for the product in the basket for this line item.

C#

```
public string ProductTitle;
```

Visual Basic

```
Public Property ProductTitle() As string
```

BasketItem.ProductType Property

Gets or sets the type of the product in the basket for this line item.

C#

```
public EkEnumeration.CatalogEntryType ProductType;
```

Visual Basic

```
Public Property ProductType() As EkEnumeration.CatalogEntryType
```

BasketItem.Quantity Property

Gets or sets the quantity of the product in the basket for this line item.

C#

```
public int Quantity;
```

Visual Basic

```
Public Property Quantity() As Integer
```

BasketItem.Quicklink Property

Gets or sets the product's Quicklink.

C#

```
public string Quicklink;
```

Visual Basic

```
Public Property Quicklink() As string
```

BasketItem.SalePrice Property

Sets the base sale price for the line item. The returned sale price will reflect any adjustments based upon kit options or product variants selected.

C#

```
public decimal SalePrice;
```

Visual Basic

```
Public Property SalePrice() As decimal
```

BasketItem.TaxClassId Property

Gets or sets the tax class Id (see page 1451) for the product in the basket for this line item.

C#

```
public long TaxClassId;
```

Visual Basic

```
Public Property TaxClassId() As long
```

BasketItem.Variant Property

Gets or sets and applicable varioant information for the product in the basket for this line item.

C#

```
public BasketVariantData Variant;
```

Visual Basic

```
Public Property Variant() As BasketVariantData
```

BasketItem.Weight Property

Gets or sets the weight of the product. Used for shipping.

C#

```
public Weight Weight;
```

Visual Basic

```
Public Property Weight() As Weight
```

BasketItemCollection Class

Collection (see page 95) class for BasketItems.

Class Hierarchy

C#

```
public class BasketItemCollection : Collection<BasketItem>;
```

Visual Basic

```
Public Class BasketItemCollection
Inherits Collection
```

File

BasketItem.cs

BasketItemCollection.BasketItemCollection Constructor

C#

```
public BasketItemCollection();
```

Visual Basic

```
Public Sub New()
```

Description

This is BasketItemCollection, a member of class BasketItemCollection.

BasketItemCollection Methods

The methods of the BasketItemCollection class are listed here.

Public Methods

	Name	Description
	AddRange (see page 1455)	This is AddRange, a member of class BasketItemCollection.
	Find (see page 1455)	This is Find, a member of class BasketItemCollection.
	FindAll (see page 1455)	This is FindAll, a member of class BasketItemCollection.
	FindIndex (see page 1455)	This is FindIndex, a member of class BasketItemCollection.

BasketItemCollection.AddRange Method

C#

```
public void AddRange(IEnumerable<BasketItem> collection);
```

Visual Basic

```
Public Function AddRange(collection As IEnumerable<BasketItem>) As void
```

Description

This is AddRange, a member of class BasketItemCollection.

BasketItemCollection.Find Method

C#

```
public BasketItem Find(Predicate<BasketItem> match);
```

Visual Basic

```
Public Function Find(match As Predicate<BasketItem>) As BasketItem
```

Description

This is Find, a member of class BasketItemCollection.

BasketItemCollection.FindAll Method

C#

```
public List<BasketItem> FindAll(Predicate<BasketItem> match);
```

Visual Basic

```
Public Function FindAll(match As Predicate<BasketItem>) As List<BasketItem>
```

Description

This is FindAll, a member of class BasketItemCollection.

BasketItemCollection.FindIndex Method

C#

```
public int FindIndex(Predicate<BasketItem> match);
```

Visual Basic

```
Public Function FindIndex(match As Predicate<BasketItem>) As Integer
```

Description

This is FindIndex, a member of class BasketItemCollection.

BasketVariantData Class

Class Hierarchy

Ektron.Cms.Commerce.BasketVariantData

C#

```
public class BasketVariantData;
```

Visual Basic

```
Public Class BasketVariantData
```

File

VariantData.cs

BasketVariantData.BasketVariantData Constructor

C#

```
public BasketVariantData();
```

Visual Basic

```
Public Sub New()
```

Description

This is BasketVariantData, a member of class BasketVariantData.

BasketVariantData Properties

The properties of the BasketVariantData class are listed here.

Public Properties

	Name	Description
	Dimensions (see page 1456)	Gets or sets the dimensions of the product. Used for shipping.
	Id (see page 1456)	This is Id, a member of class BasketVariantData.
	Language (see page 1457)	This is Language, a member of class BasketVariantData.
	ListPrice (see page 1457)	This is ListPrice, a member of class BasketVariantData.
	Pricing (see page 1457)	The variant pricing, including currencies and quantity.
	SalePrice (see page 1457)	This is SalePrice, a member of class BasketVariantData.
	Sku (see page 1457)	This is Sku, a member of class BasketVariantData.
	TaxClassId (see page 1458)	Gets or set the TaxClassId of the Variant.
	Title (see page 1458)	This is Title, a member of class BasketVariantData.
	Weight (see page 1458)	Gets or sets the weight of the product. Used for shipping.

BasketVariantData.Dimensions Property

Gets or sets the dimensions of the product. Used for shipping.

C#

```
public Dimensions Dimensions;
```

Visual Basic

```
Public Property Dimensions() As Dimensions
```

BasketVariantData.Id Property

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

Description

This is Id, a member of class BasketVariantData.

BasketVariantData.Language Property

C#

```
public long Language;
```

Visual Basic

```
Public Property Language() As long
```

Description

This is Language, a member of class BasketVariantData.

BasketVariantData.ListPrice Property

C#

```
public decimal ListPrice;
```

Visual Basic

```
Public Property ListPrice() As decimal
```

Description

This is ListPrice, a member of class BasketVariantData.

BasketVariantData.Pricing Property

The variant pricing, including currencies and quantity.

C#

```
public PricingData Pricing;
```

Visual Basic

```
Public Property Pricing() As PricingData
```

Returns

PricingData (see page 1540) of the variant.

BasketVariantData.SalePrice Property

C#

```
public decimal SalePrice;
```

Visual Basic

```
Public Property SalePrice() As decimal
```

Description

This is SalePrice, a member of class BasketVariantData.

BasketVariantData.Sku Property

C#

```
public string Sku;
```

Visual Basic

```
Public Property Sku() As string
```

Description

This is Sku, a member of class BasketVariantData.

BasketVariantData.TaxClassId Property

Gets or set the TaxClassId of the Variant.

C#

```
public long TaxClassId;
```

Visual Basic

```
Public Property TaxClassId() As long
```

BasketVariantData.Title Property

C#

```
public string Title;
```

Visual Basic

```
Public Property Title() As string
```

Description

This is Title, a member of class BasketVariantData.

BasketVariantData.Weight Property

Gets or sets the weight of the product. Used for shipping.

C#

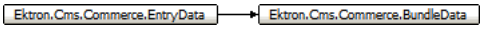
```
public Weight Weight;
```

Visual Basic

```
Public Property Weight() As Weight
```

BundleData Class

Class Hierarchy



C#

```
public class BundleData : EntryData;
```

Visual Basic

```
Public Class BundleData
Inherits EntryData
```

File

BundleData.cs

BundleData.BundleData Constructor

C#

```
public BundleData();
```

Visual Basic

```
Public Sub New()
```

Description

This is BundleData, a member of class BundleData.

BundleData Properties

The properties of the BundleData class are listed here.

Public Properties

	Name	Description
	BundledItems (see page 1459)	This is BundledItems, a member of class BundleData.

BundleData.BundledItems Property

C#

```
public List<EntryData> BundledItems;
```

Visual Basic

```
Public Property BundledItems() As List<EntryData>
```

Description

This is BundledItems, a member of class BundleData.

CatalogData Class

Class Hierarchy

Ektron.Cms.Commerce.CatalogData

C#

```
public class CatalogData;
```

Visual Basic

```
Public Class CatalogData
```

File

CatalogData.cs

CatalogData Properties

The properties of the CatalogData class are listed here.

Public Properties

	Name	Description
	Id (see page 1460)	This is Id, a member of class CatalogData.
	Name (see page 1460)	This is Name, a member of class CatalogData.

CatalogData.Id Property

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

Description

This is Id, a member of class CatalogData.

CatalogData.Name Property

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

Description

This is Name, a member of class CatalogData.

ContractData Class

Class Hierarchy

Ektron.Cms.Commerce.ContractData

C#

```
public class ContractData;
```

Visual Basic

```
Public Class ContractData
```

File

ContractData.cs

Description

This is class Ektron.Cms.Commerce.ContractData.

ContractData Constructor

This is the overview for the ContractData constructor overload.

Overload List

	Name	Description
	ContractData.ContractData () (see page 1461)	This is ContractData, a member of class ContractData.
	ContractData.ContractData (string) (see page 1461)	This is ContractData, a member of class ContractData.

ContractData.ContractData Constructor ()

C#

```
public ContractData();
```

Visual Basic

```
Public Sub New()
```

Description

This is ContractData, a member of class ContractData.

ContractData.ContractData Constructor (string)

C#

```
public ContractData(string contractName);
```

Visual Basic

```
Public Sub New(contractName As string)
```

Description

This is ContractData, a member of class ContractData.

ContractData Properties

The properties of the ContractData class are listed here.

Public Properties

	Name	Description
	Id (see page 1461)	Gets or sets the Id of the contract.
	Name (see page 1461)	Gets or sets the contract name.

ContractData.Id Property

Gets or sets the Id of the contract.

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

ContractData.Name Property

Gets or sets the contract name.

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

CountryData Class

Gets or sets information about a specified country.

Class Hierarchy

Ektron.Cms.Commerce.CountryData

C#

```
public class CountryData;
```

Visual Basic

```
Public Class CountryData
```

File

CountryData.cs

CountryData Constructor

This is the overview for the CountryData constructor overload.

Overload List

	Name	Description
	CountryData.CountryData () (see page 1462)	This is CountryData, a member of class CountryData.
	CountryData.CountryData (int, string, string, string, bool) (see page 1462)	

CountryData.CountryData Constructor ()

C#

```
public CountryData();
```

Visual Basic

```
Public Sub New()
```

Description

This is CountryData, a member of class CountryData.

CountryData.CountryData Constructor (int, string, string, string, bool)

C#

```
public CountryData(int id, string name, string shortIsoCode, string longIsoCode, bool enabled);
```

Visual Basic

```
Public Sub New(id As Integer, name As string, shortIsoCode As string, longIsoCode As string,
enabled As bool)
```

Parameters

Parameters	Description
int id	The ID of the country. The ID is the ISO numeric code of the country.
string name	The name of the country.
string shortIsoCode	The 2 letter ISO code of the country.
string longIsoCode	The 3 letter ISO code of the country.
bool enabled	Indicates whether or not the country is enabled for use.

CountryData Methods

The methods of the CountryData class are listed here.

Public Methods

	Name	Description
	ToString (see page 1463)	Returns Name (see page 1464) of Country.
	Validate (see page 1463)	Validates the CountryData (see page 1461) object is valid for saving.

CountryData.ToString Method

Returns Name (see page 1464) of Country.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

CountryData.Validate Method

Validates the CountryData (see page 1461) object is valid for saving.

C#

```
public ValidationResults Validate();
```

Visual Basic

```
Public Function Validate() As ValidationResults
```

Returns

Returns a ValidationResult object with any validation errors.

CountryData Properties

The properties of the CountryData class are listed here.

Public Properties

	Name	Description
	Enabled (see page 1463)	Gets or sets whether or not the country is enabled for use.
	Id (see page 1464)	Gets or sets the ID of the country. The ID is the ISO numeric code of the country. This property is required when creating an country data object.
	LongIsoCode (see page 1464)	Gets or sets the 3 letter ISO code of the country. This property is required when creating an country data object.
	Name (see page 1464)	Gets or sets the name of the country. This property is required when creating an country data object.
	ShortIsoCode (see page 1464)	Gets or sets the 2 letter ISO code of the country. This property is required when creating an country data object.

CountryData.Enabled Property

Gets or sets whether or not the country is enabled for use.

C#

```
public bool Enabled;
```

Visual Basic

```
Public Property Enabled() As bool
```

CountryData.Id Property

Gets or sets the ID of the country. The ID is the ISO numeric code of the country. This property is required when creating an country data object.

C#

```
public int Id;
```

Visual Basic

```
Public Property Id() As Integer
```

CountryData.LongIsoCode Property

Gets or sets the 3 letter ISO code of the country. This property is required when creating an country data object.

C#

```
public string LongIsoCode;
```

Visual Basic

```
Public Property LongIsoCode() As string
```

CountryData.Name Property

Gets or sets the name of the country. This property is required when creating an country data object.

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

CountryData.ShortIsoCode Property

Gets or sets the 2 letter ISO code of the country. This property is required when creating an country data object.

C#

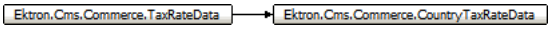
```
public string ShortIsoCode;
```

Visual Basic

```
Public Property ShortIsoCode() As string
```

CountryTaxRateData Class

Class Hierarchy



C#

```
public class CountryTaxRateData : TaxRateData;
```

Visual Basic

```
Public Class CountryTaxRateData
Inherits TaxRateData
```

File

CountryTaxRateData.cs

Description

This is class Ektron.Cms.Commerce.CountryTaxRateData.

CountryTaxRateData Constructor

This is the overview for the CountryTaxRateData constructor overload.

Overload List

	Name	Description
	CountryTaxRateData.CountryTaxRateData () (see page 1465)	This is CountryTaxRateData, a member of class CountryTaxRateData.
	CountryTaxRateData.CountryTaxRateData (long, long, decimal) (see page 1465)	This is CountryTaxRateData, a member of class CountryTaxRateData.
	CountryTaxRateData.CountryTaxRateData (long, long, decimal, long) (see page 1465)	This is CountryTaxRateData, a member of class CountryTaxRateData.

CountryTaxRateData.CountryTaxRateData Constructor ()

C#

```
public CountryTaxRateData();
```

Visual Basic

```
Public Sub New()
```

Description

This is CountryTaxRateData, a member of class CountryTaxRateData.

CountryTaxRateData.CountryTaxRateData Constructor (long, long, decimal)

C#

```
public CountryTaxRateData(long countryId, long taxClassId, decimal rate);
```

Visual Basic

```
Public Sub New(countryId As long, taxClassId As long, rate As decimal)
```

Description

This is CountryTaxRateData, a member of class CountryTaxRateData.

CountryTaxRateData.CountryTaxRateData Constructor (long, long, decimal, long)

C#

```
public CountryTaxRateData(long countryId, long taxClassId, decimal rate, long id);
```

Visual Basic

```
Public Sub New(countryId As long, taxClassId As long, rate As decimal, id As long)
```

Description

This is CountryTaxRateData, a member of class CountryTaxRateData.

CountryTaxRateData Properties

The properties of the CountryTaxRateData class are listed here.

Public Properties

	Name	Description
	CountryData (see page 1466)	Gets or sets the CountryData object associated with this tax rate.

CountryTaxRateData.CountryData Property

Gets or sets the CountryData object associated with this tax rate.

C#

```
public CountryData CountryData;
```

Visual Basic

```
Public Property CountryData() As CountryData
```

CouponData Class

Class Hierarchy

Ektron.Cms.Commerce.CouponData

C#

```
public class CouponData;
```

Visual Basic

```
Public Class CouponData
```

File

CouponData.cs

Description

This is class Ektron.Cms.Commerce.CouponData.

CouponData.CouponData Constructor

C#

```
public CouponData();
```

Visual Basic

```
Public Sub New()
```

Description

This is CouponData, a member of class CouponData.

CouponData Properties

The properties of the CouponData class are listed here.

Public Properties

	Name	Description
	Code (see page 1467)	Gets or sets the coupon code.
	CouponType (see page 1467)	Gets or sets the coupon type.
	CurrencyId (see page 1467)	Gets or sets the Id (see page 1468) of the currency this coupon applies to.
	Description (see page 1468)	Gets or sets the description of the coupon.
	DiscountType (see page 1468)	Gets or sets the Coupon Discount Type.
	DiscountValue (see page 1468)	Gets or sets the value of the discount. For a fixed amount, this would be the amount (i.e. 5.00). For percentage, it would be the discount percent value in whole numbers (i.e. 15% = 15.00).
	ExpirationDate (see page 1468)	Gets or sets the date the coupon expires.
	Id (see page 1468)	Gets or sets the Id of the coupon.
	IsActive (see page 1468)	Determines whether or not this coupon is currently active. Deleted coupons remain in the system and are marked inactive.
	IsCombinable (see page 1469)	Determins whether or not this coupon can be combined with other coupons. If true, this coupon can be combined with other "Combinable" coupons.
	IsRedeemable (see page 1469)	Determines whether or not this coupon is currently redeemable. A coupon may not be reddemable if the it is inactive, not within its valid period, or has been redeemed the maximum number of times.
	MaximumAmount (see page 1469)	Gets or sets the maximum order amount that this coupon can be applied to.
	MaximumUses (see page 1469)	Gets or sets the maximim number of times this coupon can be used before it expires.
	MinimumAmount (see page 1469)	Gets or sets the minimum order amount required before the coupon is applicable.
	OnePerCustomer (see page 1469)	Gets or sets the OnePerCuster value. If true, this coupon can only be used once by each customer.
	StartDate (see page 1470)	Gets or sets the date the coupon becomes valid.

CouponData.Code Property

Gets or sets the coupon code.

C#

```
public string Code;
```

Visual Basic

```
Public Property Code() As string
```

CouponData.CouponType Property

Gets or sets the coupon type.

C#

```
public EkEnumeration.CouponType CouponType;
```

Visual Basic

```
Public Property CouponType() As EkEnumeration.CouponType
```

CouponData.CurrencyId Property

Gets or sets the Id (see page 1468) of the currency this coupon applies to.

C#

```
public int CurrencyId;
```

Visual Basic

```
Public Property CurrencyId() As Integer
```

CouponData.Description Property

Gets or sets the description of the coupon.

C#

```
public string Description;
```

Visual Basic

```
Public Property Description() As string
```

CouponData.DiscountType Property

Gets or sets the Coupon Discount Type.

C#

```
public EkEnumeration.CouponDiscountType DiscountType;
```

Visual Basic

```
Public Property DiscountType() As EkEnumeration.CouponDiscountType
```

CouponData.DiscountValue Property

Gets or sets the value of the discount. For a fixed amount, this would be the amount (i.e. 5.00). For percentage, it would be the discount percent value in whole numbers (i.e. 15% = 15.00).

C#

```
public decimal DiscountValue;
```

Visual Basic

```
Public Property DiscountValue() As decimal
```

CouponData.ExpirationDate Property

Gets or sets the date the coupon expires.

C#

```
public DateTime ExpirationDate;
```

Visual Basic

```
Public Property ExpirationDate() As DateTime
```

CouponData.Id Property

Gets or sets the Id of the coupon.

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

CouponData.IsActive Property

Determines whether or not this coupon is currently active. Deleted coupons remain in the system and are marked inactive.

C#

```
public bool IsActive;
```

Visual Basic

```
Public Property IsActive() As bool
```

CouponData.IsCombinable Property

Determins whether or not this coupon can be combined with other coupons. If true, this coupon can be combined with other "Combinable" coupons.

C#

```
public bool IsCombinable;
```

Visual Basic

```
Public Property IsCombinable() As bool
```

CouponData.IsRedeemable Property

Determines whether or not this coupon is currently redeemable. A coupon may not be reddemable if the it is inactive, not within its valid period, or has been redeemed the maximum number of times.

C#

```
public bool IsRedeemable;
```

Visual Basic

```
Public Property IsRedeemable() As bool
```

CouponData.MaximumAmount Property

Gets or sets the maximum order amount that this coupon can be applied to.

C#

```
public decimal MaximumAmount;
```

Visual Basic

```
Public Property MaximumAmount() As decimal
```

CouponData.MaximumUses Property

Gets or sets the maximim number of times this coupon can be used before it expires.

C#

```
public int MaximumUses;
```

Visual Basic

```
Public Property MaximumUses() As Integer
```

CouponData.MinimumAmount Property

Gets or sets the minimum order amount required before the coupon is applicable.

C#

```
public decimal MinimumAmount;
```

Visual Basic

```
Public Property MinimumAmount() As decimal
```

CouponData.OnePerCustomer Property

Gets or sets the OnePerCuster value. If true, this coupon can only be used once by each customer.

C#

```
public bool OnePerCustomer;
```

Visual Basic

```
Public Property OnePerCustomer() As bool
```

CouponData.StartDate Property

Gets or sets the date the coupon becomes valid.

C#

```
public DateTime StartDate;
```

Visual Basic

```
Public Property StartDate() As DateTime
```

CouponEntryData Class

Class Hierarchy

Ektron.Cms.Commerce.CouponEntryData

C#

```
public class CouponEntryData;
```

Visual Basic

```
Public Class CouponEntryData
```

File

CouponEntryData.cs

Description

This is class Ektron.Cms.Commerce.CouponEntryData.

CouponEntryData Constructor

This is the overview for the CouponEntryData constructor overload.

Overload List

	Name	Description
	CouponEntryData.CouponEntryData () (see page 1470)	This is CouponEntryData, a member of class CouponEntryData.
	CouponEntryData.CouponEntryData (long, EkEnumeration.CMSObjectTypes, long, string, bool) (see page 1471)	This is CouponEntryData, a member of class CouponEntryData.

CouponEntryData.CouponEntryData Constructor ()

C#

```
public CouponEntryData();
```

Visual Basic

```
Public Sub New()
```

Description

This is CouponEntryData, a member of class CouponEntryData.

CouponEntryData.CouponEntryData Constructor (long, EkEnumeration.CMSObjectTypes, long, string, bool)

C#

```
public CouponEntryData(long objectId, EkEnumeration.CMSObjectTypes objectType, long couponId,
string title, bool isIncluded);
```

Visual Basic

```
Public Sub New(objectId As long, objectType As EkEnumeration.CMSObjectTypes, couponId As long,
title As string, isIncluded As bool)
```

Description

This is CouponEntryData, a member of class CouponEntryData.

CouponEntryData Properties

The properties of the CouponEntryData class are listed here.

Public Properties

	Name	Description
	CouponId (see page 1471)	Gets or sets the Id of the coupon
	DataState (see page 1471)	Gets or sets the current saved state for the CouponEntry. Used by the Cms (see page 7) internally.
	IsIncluded (see page 1472)	Gets or sets whether or not the Object is included in the coupon. If true, the object is included in the coupon. If false, the coupon is NOT included in the coupon.
	ObjectId (see page 1472)	Gets or sets the Id of object the coupon is applied to.
	ObjectType (see page 1472)	Gets or sets the type of object this coupon is applied to.
	Path (see page 1472)	Gets or sets the Path of the object.
	Title (see page 1472)	Gets or sets the Title of the object.

CouponEntryData.CouponId Property

Gets or sets the Id of the coupon

C#

```
public long CouponId;
```

Visual Basic

```
Public Property CouponId() As long
```

CouponEntryData.DataState Property

Gets or sets the current saved state for the CouponEntry. Used by the Cms (see page 7) internally.

C#

```
public EkEnumeration.DataState DataState;
```

Visual Basic

```
Public Property DataState() As EkEnumeration.DataState
```

CouponEntryData.IsIncluded Property

Gets or sets whether or not the Object is included in the coupon. If true, the object is included in the coupon. If false, the coupon is NOT included in the coupon.

C#

```
public bool IsIncluded;
```

Visual Basic

```
Public Property IsIncluded() As bool
```

CouponEntryData.ObjectId Property

Gets or sets the Id of object the coupon is applied to.

C#

```
public long ObjectId;
```

Visual Basic

```
Public Property ObjectId() As long
```

CouponEntryData.ObjectType Property

Gets or sets the type of object this coupon is applied to.

C#

```
public EkEnumeration.CMSObjectTypes ObjectType;
```

Visual Basic

```
Public Property ObjectType() As EkEnumeration.CMSObjectTypes
```

CouponEntryData.Path Property

Gets or sets the Path of the object.

C#

```
public string Path;
```

Visual Basic

```
Public Property Path() As string
```

CouponEntryData.Title Property

Gets or sets the Title of the object.

C#

```
public string Title;
```

Visual Basic

```
Public Property Title() As string
```

CurrencyData Class

Data class representing currencies in system.

Class Hierarchy

Ektron.Cms.Commerce.CurrencyData

C#

```
public class CurrencyData;
```

Visual Basic

```
Public Class CurrencyData
```

File

CurrencyData.cs

CurrencyData Properties

The properties of the CurrencyData class are listed here.

Public Properties

	Name	Description
	AlphaIsoCode (see page 1473)	Gets or sets the 3 letter ISO code of the currency.
	CultureCode (see page 1473)	Gets or sets the culture code associated with this currency. This is used for formatting currency values for this currency.
	CurrencySymbol (see page 1473)	Read only function that gets the symbol for this currency.
	Enabled (see page 1474)	Gets or sets whether or not the currency is enabled for use.
	Id (see page 1474)	Gets or sets the Id of the currency. The Id is the ISO numeric code of the currency.
	ISOCurrencySymbol (see page 1474)	Read only function that gets the ISO symbol for this currency.
	Name (see page 1474)	Gets or sets the name of the currency.

CurrencyData.AlphaIsoCode Property

Gets or sets the 3 letter ISO code of the currency.

C#

```
public string AlphaIsoCode;
```

Visual Basic

```
Public Property AlphaIsoCode() As string
```

CurrencyData.CultureCode Property

Gets or sets the culture code associated with this currency. This is used for formatting currency values for this currency.

C#

```
public string CultureCode;
```

Visual Basic

```
Public Property CultureCode() As string
```

CurrencyData.CurrencySymbol Property

Read only function that gets the symbol for this currency.

C#

```
public string CurrencySymbol;
```

Visual Basic

```
Public Property CurrencySymbol() As string
```

Example

$

CurrencyData.Enabled Property

Gets or sets whether or not the currency is enabled for use.

C#

```
public bool Enabled;
```

Visual Basic

```
Public Property Enabled() As bool
```

CurrencyData.Id Property

Gets or sets the Id of the currency. The Id is the ISO numeric code of the currency.

C#

```
public int Id;
```

Visual Basic

```
Public Property Id() As Integer
```

CurrencyData.ISOCurrencySymbol Property

Read only function that gets the ISO symbol for this currency.

C#

```
public string ISOCurrencySymbol;
```

Visual Basic

```
Public Property ISOCurrencySymbol() As string
```

Example

USD

CurrencyData.Name Property

Gets or sets the name of the currency.

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

CurrencyPricingData Class

Represents Pricing Information for agiven currency.

Class Hierarchy

Ektron.Cms.Commerce.CurrencyPricingData

C#

```
public class CurrencyPricingData;
```

Visual Basic

```
Public Class CurrencyPricingData
```

File

CurrencyPricingData.cs

CurrencyPricingData Constructor

This is the overview for the CurrencyPricingData constructor overload.

Overload List

	Name	Description
	CurrencyPricingData.CurrencyPricingData () (see page 1475)	This is CurrencyPricingData, a member of class CurrencyPricingData.
	CurrencyPricingData.CurrencyPricingData (long, decimal, decimal) (see page 1475)	Creates a CurrencyPricingData with a single tierprice of quantity 1 and the supplied sale price.
	CurrencyPricingData.CurrencyPricingData (long, decimal, EkEnumeration.PricingType) (see page 1475)	This is CurrencyPricingData, a member of class CurrencyPricingData.

CurrencyPricingData.CurrencyPricingData Constructor ()

C#

```
public CurrencyPricingData();
```

Visual Basic

```
Public Sub New()
```

Description

This is CurrencyPricingData, a member of class CurrencyPricingData.

CurrencyPricingData.CurrencyPricingData Constructor (long, decimal, EkEnumeration.PricingType)

C#

```
public CurrencyPricingData(long currencyId, decimal listPrice, EkEnumeration.PricingType
pricingType);
```

Visual Basic

```
Public Sub New(currencyId As long, listPrice As decimal, pricingType As
EkEnumeration.PricingType)
```

Description

This is CurrencyPricingData, a member of class CurrencyPricingData.

CurrencyPricingData.CurrencyPricingData Constructor (long, decimal, decimal)

Creates a CurrencyPricingData with a single tierprice of quantity 1 and the supplied sale price.

C#

```
public CurrencyPricingData(long currencyId, decimal listPrice, decimal salePrice);
```

Visual Basic

```
Public Sub New(currencyId As long, listPrice As decimal, salePrice As decimal)
```

Parameters

Parameters	Description
long currencyId	Id of currency to create pricing for.
decimal listPrice	List Price for currency.
decimal salePrice	sale Price for single tier item.

CurrencyPricingData Methods

The methods of the CurrencyPricingData class are listed here.

Public Methods

	Name	Description
	Clone (see page 1476)	This is Clone, a member of class CurrencyPricingData.
	GetSalePrice (see page 1476)	This is GetSalePrice, a member of class CurrencyPricingData.
	Validate (see page 1476)	This is Validate, a member of class CurrencyPricingData.

CurrencyPricingData.Clone Method

C#

```
public CurrencyPricingData Clone();
```

Visual Basic

```
Public Function Clone() As CurrencyPricingData
```

Description

This is Clone, a member of class CurrencyPricingData.

CurrencyPricingData.GetSalePrice Method

C#

```
public decimal GetSalePrice(int quantity);
```

Visual Basic

```
Public Function GetSalePrice(quantity As Integer) As decimal
```

Description

This is GetSalePrice, a member of class CurrencyPricingData.

CurrencyPricingData.Validate Method

C#

```
public ValidationResults Validate();
```

Visual Basic

```
Public Function Validate() As ValidationResults
```

Description

This is Validate, a member of class CurrencyPricingData.

CurrencyPricingData Properties

The properties of the CurrencyPricingData class are listed here.

Public Properties

	Name	Description
	AlphaIsoCode (see page 1477)	Gets or sets the alpha iso code of the currency associated with these prices.
	CurrencyId (see page 1477)	Gets or sets the Id of currency that this pricing represents.
	ListPrice (see page 1477)	Gets or sets the list price for this currency.
	NumericIsoCode (see page 1477)	Gets or sets the numeric iso code of the currency associated with these prices.
	PricingType (see page 1478)	Gets or sets the PricingType forthe currency.
	SalePrice (see page 1478)	Gets the default sale price for a single quantity.
	TierPrices (see page 1478)	Gets all the tier prices for this currency.

CurrencyPricingData.AlphaIsoCode Property

Gets or sets the alpha iso code of the currency associated with these prices.

C#

```
public string AlphaIsoCode;
```

Visual Basic

```
Public Property AlphaIsoCode() As string
```

CurrencyPricingData.CurrencyId Property

Gets or sets the Id of currency that this pricing represents.

C#

```
public long CurrencyId;
```

Visual Basic

```
Public Property CurrencyId() As long
```

CurrencyPricingData.ListPrice Property

Gets or sets the list price for this currency.

C#

```
public decimal ListPrice;
```

Visual Basic

```
Public Property ListPrice() As decimal
```

CurrencyPricingData.NumericIsoCode Property

Gets or sets the numeric iso code of the currency associated with these prices.

C#

```
public int NumericIsoCode;
```

Visual Basic

```
Public Property NumericIsoCode() As Integer
```

CurrencyPricingData.PricingType Property

Gets or sets the PricingType forthe currency.

C#

```
public Common.EkEnumeration.PricingType PricingType;
```

Visual Basic

```
Public Property PricingType() As Common.EkEnumeration.PricingType
```

CurrencyPricingData.SalePrice Property

Gets the default sale price for a single quantity.

C#

```
public decimal SalePrice;
```

Visual Basic

```
Public Property SalePrice() As decimal
```

CurrencyPricingData.TierPrices Property

Gets all the tier prices for this currency.

C#

```
public List<TierPriceData> TierPrices;
```

Visual Basic

```
Public Property TierPrices() As List<TierPriceData>
```

CustomerData Class

Data class representing Customers in system.

Class Hierarchy

Ektron.Cms.Commerce.CustomerData

C#

```
public class CustomerData;
```

Visual Basic

```
Public Class CustomerData
```

File

CustomerData.cs

CustomerData Properties

The properties of the CustomerData class are listed here.

Public Properties

	Name	Description
	AverageOrderValue (see page 1479)	Gets the average order placed by this customer.
	BillingAddressId (see page 1479)	Gets or sets the Address Id (see page 1480) of the billing address for the customer.
	DisplayName (see page 1479)	Gets or sets the display name of the customer.
	EmailAddress (see page 1479)	Gets or sets the email address of the customer.
	FirstName (see page 1480)	Gets or sets the first name of the customer.
	Id (see page 1480)	Gets or sets the Id of the customer.
	IsDeleted (see page 1480)	Gets the IsDeleted status of the customer.
	IsMembershipUser (see page 1480)	Gets or sets the IsMembershipUser status of the customer.
	LastName (see page 1480)	Gets or sets the last of the customer.
	Password (see page 1480)	Gets or sets the password for the customer.
	ShippingAddressId (see page 1481)	Gets or sets Address Id (see page 1480) of the shipping address for this customer.
	TotalOrders (see page 1481)	Gets or sets the toal number of orders placed by the customer.
	TotalOrderValue (see page 1481)	Gets or sets the total value of orders placed by this customer.
	UserId (see page 1481)	Gets or sets the Id (see page 1480) of the Cms (see page 7) User (see page 950) object associated with this customer.
	UserName (see page 1481)	Gets or sets the username for the customer.

CustomerData.AverageOrderValue Property

Gets the average order placed by this customer.

C#

```
public decimal AverageOrderValue;
```

Visual Basic

```
Public Property AverageOrderValue() As decimal
```

CustomerData.BillingAddressId Property

Gets or sets the Address Id (see page 1480) of the billing address for the customer.

C#

```
public long BillingAddressId;
```

Visual Basic

```
Public Property BillingAddressId() As long
```

CustomerData.DisplayName Property

Gets or sets the display name of the customer.

C#

```
public string DisplayName;
```

Visual Basic

```
Public Property DisplayName() As string
```

CustomerData.EmailAddress Property

Gets or sets the email address of the customer.

C#

```
public string EmailAddress;
```

Visual Basic

```
Public Property EmailAddress() As string
```

CustomerData.FirstName Property

Gets or sets the first name of the customer.

C#

```
public string FirstName;
```

Visual Basic

```
Public Property FirstName() As string
```

CustomerData.Id Property

Gets or sets the Id of the customer.

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

CustomerData.IsDeleted Property

Gets the IsDeleted status of the customer.

C#

```
public bool IsDeleted;
```

Visual Basic

```
Public Property IsDeleted() As bool
```

CustomerData.IsMembershipUser Property

Gets or sets the IsMembershipUser status of the customer.

C#

```
public bool IsMembershipUser;
```

Visual Basic

```
Public Property IsMembershipUser() As bool
```

CustomerData.LastName Property

Gets or sets the last of the customer.

C#

```
public string LastName;
```

Visual Basic

```
Public Property LastName() As string
```

CustomerData.Password Property

Gets or sets the password for the customer.

C#

```
public string Password;
```

Visual Basic

```
Public Property Password() As string
```

CustomerData.ShippingAddressId Property

Gets or sets Address Id (see page 1480) of the shipping address for this customer.

C#

```
public long ShippingAddressId;
```

Visual Basic

```
Public Property ShippingAddressId() As long
```

CustomerData.TotalOrders Property

Gets or sets the toal number of orders placed by the customer.

C#

```
public int TotalOrders;
```

Visual Basic

```
Public Property TotalOrders() As Integer
```

CustomerData.TotalOrderValue Property

Gets or sets the total value of orders placed by this customer.

C#

```
public decimal TotalOrderValue;
```

Visual Basic

```
Public Property TotalOrderValue() As decimal
```

CustomerData.UserId Property

Gets or sets the Id (see page 1480) of the Cms (see page 7) User (see page 950) object associated with this customer.

C#

```
public long UserId;
```

Visual Basic

```
Public Property UserId() As long
```

CustomerData.UserName Property

Gets or sets the username for the customer.

C#

```
public string UserName;
```

Visual Basic

```
Public Property UserName() As string
```

Dimensions Class

Class used to define dimension information.

Class Hierarchy

C#

```
public class Dimensions : IComparable<Dimensions>;
```

Visual Basic

```
Public Class Dimensions
Inherits IComparable
```

File

Measure.cs

Dimensions Constructor

This is the overview for the Dimensions constructor overload.

Overload List

	Name	Description
	Dimensions.Dimensions () (see page 1482)	This is Dimensions, a member of class Dimensions.
	Dimensions.Dimensions (LinearUnit, float, float, float) (see page 1482)	

Dimensions.Dimensions Constructor ()

C#

```
public Dimensions();
```

Visual Basic

```
Public Sub New()
```

Description

This is Dimensions, a member of class Dimensions.

Dimensions.Dimensions Constructor (LinearUnit, float, float, float)

C#

```
public Dimensions(LinearUnit units, float length, float width, float height);
```

Visual Basic

```
Public Sub New(units As LinearUnit, length As float, width As float, height As float)
```

Parameters

Parameters	Description
LinearUnit units	Linear Units (see page 1484) that define the dimensions.

float length	Length (see page 1484) of the package in Units (see page 1484)
float width	Length (see page 1484) of the package in Units (see page 1484)
float height	Length (see page 1484) of the package in Units (see page 1484)

Dimensions Methods

The methods of the Dimensions class are listed here.

Public Methods

	Name	Description
	Clone (see page 1483)	This is Clone, a member of class Dimensions.
	CompareTo (see page 1483)	Compares DImensions based upon CubicUnits (see page 1484)
	ConvertUnits (see page 1483)	Converts the current Dimensions (see page 1482) object to a new object in the supplied units.

Dimensions.Clone Method

C#

```
public Dimensions Clone();
```

Visual Basic

```
Public Function Clone() As Dimensions
```

Description

This is Clone, a member of class Dimensions.

Dimensions.CompareTo Method

Compares DImensions based upon CubicUnits (see page 1484)

C#

```
public int CompareTo(Dimensions other);
```

Visual Basic

```
Public Function CompareTo(other As Dimensions) As Integer
```

Parameters

Parameters	Description
Dimensions other	

Dimensions.ConvertUnits Method

Converts the current Dimensions (see page 1482) object to a new object in the supplied units.

C#

```
public Dimensions ConvertUnits(LinearUnit toUnit);
```

Visual Basic

```
Public Function ConvertUnits(toUnit As LinearUnit) As Dimensions
```

Parameters

Parameters	Description
LinearUnit toUnit	LinearUnit (see page 1585) to convert to

Dimensions Properties

The properties of the Dimensions class are listed here.

Public Properties

	Name	Description
	CubicUnits (see page 1484)	Gets the dimensions cubic units.
	Height (see page 1484)	Gets or sets the Length (see page 1484) of the package in Units (see page 1484)
	Length (see page 1484)	Gets or sets the Length of the package in Units (see page 1484).
	Units (see page 1484)	Gets or sets the Linear Units that define the dimensions.
	Width (see page 1484)	Gets or sets the Length (see page 1484) of the package in Units (see page 1484)

Dimensions.CubicUnits Property

Gets the dimensions cubic units.

C#

```
public float CubicUnits;
```

Visual Basic

```
Public Property CubicUnits() As float
```

Dimensions.Height Property

Gets or sets the Length (see page 1484) of the package in Units (see page 1484)

C#

```
public float Height;
```

Visual Basic

```
Public Property Height() As float
```

Dimensions.Length Property

Gets or sets the Length of the package in Units (see page 1484).

C#

```
public float Length;
```

Visual Basic

```
Public Property Length() As float
```

Dimensions.Units Property

Gets or sets the Linear Units that define the dimensions.

C#

```
public LinearUnit Units;
```

Visual Basic

```
Public Property Units() As LinearUnit
```

Dimensions.Width Property

Gets or sets the Length (see page 1484) of the package in Units (see page 1484)

C#

```
public float Width;
```

Visual Basic

```
Public Property Width() As float
```

EntryAttributeData Class

Class Hierarchy

C#

```
public class EntryAttributeData : ProductTypeAttributeData;
```

Visual Basic

```
Public Class EntryAttributeData
Inherits ProductTypeAttributeData
```

File

ProductAttributeData.cs

Description

This is class Ektron.Cms.Commerce.EntryAttributeData.

EntryAttributeData Constructor

This is the overview for the EntryAttributeData constructor overload.

Overload List

	Name	Description
	EntryAttributeData.EntryAttributeData () (see page 1485)	This is EntryAttributeData, a member of class EntryAttributeData.
	EntryAttributeData.EntryAttributeData (string, int, EkEnumeration.ProductTypeAttributeDataType, object, EkEnumeration.ProductTypeAttributeStatus) (see page 1486)	This is EntryAttributeData, a member of class EntryAttributeData.
	EntryAttributeData.EntryAttributeData (string, long, object, int, EkEnumeration.ProductTypeAttributeDataType, string, EkEnumeration.ProductTypeAttributeStatus) (see page 1486)	This is EntryAttributeData, a member of class EntryAttributeData.

EntryAttributeData.EntryAttributeData Constructor ()

C#

```
public EntryAttributeData();
```

Visual Basic

```
Public Sub New()
```

Description

This is EntryAttributeData, a member of class EntryAttributeData.

EntryAttributeData.EntryAttributeData Constructor (string, int, EkEnumeration.ProductTypeAttributeDataType, object, EkEnumeration.ProductTypeAttributeStatus)

C#

```
private EntryAttributeData(string name, int displayOrder,
EkEnumeration.ProductTypeAttributeDataType dataType, object defaultValue,
EkEnumeration.ProductTypeAttributeStatus activeStatus);
```

Visual Basic

```
Private Sub New(name As string, displayOrder As Integer, dataType As
EkEnumeration.ProductTypeAttributeDataType, defaultValue As object, activeStatus As
EkEnumeration.ProductTypeAttributeStatus)
```

Description

This is EntryAttributeData, a member of class EntryAttributeData.

EntryAttributeData.EntryAttributeData Constructor (string, long, object, int, EkEnumeration.ProductTypeAttributeDataType, string, EkEnumeration.ProductTypeAttributeStatus)

C#

```
public EntryAttributeData(string name, long attributeTypeId, object currentValue, int
displayOrder, EkEnumeration.ProductTypeAttributeDataType dataType, string defaultValue,
EkEnumeration.ProductTypeAttributeStatus activeStatus);
```

Visual Basic

```
Public Sub New(name As string, attributeTypeId As long, currentValue As object, displayOrder
As Integer, dataType As EkEnumeration.ProductTypeAttributeDataType, defaultValue As string,
activeStatus As EkEnumeration.ProductTypeAttributeStatus)
```

Description

This is EntryAttributeData, a member of class EntryAttributeData.

EntryAttributeData Properties

The properties of the EntryAttributeData class are listed here.

Public Properties

	Name	Description
	AttributeTypeId (see page 1487)	This is AttributeTypeId, a member of class EntryAttributeData.
	CurrentValue (see page 1487)	This is CurrentValue, a member of class EntryAttributeData.
	VersionId (see page 1487)	This is VersionId, a member of class EntryAttributeData.

EntryAttributeData.AttributeTypeId Property

C#

```
public long AttributeTypeId;
```

Visual Basic

```
Public Property AttributeTypeId() As long
```

Description

This is AttributeTypeId, a member of class EntryAttributeData.

EntryAttributeData.CurrentValue Property

C#

```
public object CurrentValue;
```

Visual Basic

```
Public Property CurrentValue() As object
```

Description

This is CurrentValue, a member of class EntryAttributeData.

EntryAttributeData.VersionId Property

C#

```
public long VersionId;
```

Visual Basic

```
Public Property VersionId() As long
```

Description

This is VersionId, a member of class EntryAttributeData.

EntryData Class

Base class for catalog entry data classes.

Class Hierarchy

C#

```
public abstract class EntryData;
```

Visual Basic

```
Public abstract Class EntryData
```

File

EntryData.cs

EntryData Constructor

This is the overview for the EntryData constructor overload.

Overload List

	Name	Description
	EntryData.EntryData () (see page 1488)	This is EntryData, a member of class EntryData.
	EntryData.EntryData (EkEnumeration.CatalogEntryType) (see page 1488)	This is EntryData, a member of class EntryData.
	EntryData.EntryData (int, EkEnumeration.CatalogEntryType) (see page 1488)	This is EntryData, a member of class EntryData.

EntryData.EntryData Constructor ()

C#

```
public EntryData();
```

Visual Basic

```
Public Sub New()
```

Description

This is EntryData, a member of class EntryData.

EntryData.EntryData Constructor (EkEnumeration.CatalogEntryType)

C#

```
public EntryData(EkEnumeration.CatalogEntryType entryType);
```

Visual Basic

```
Public Sub New(entryType As EkEnumeration.CatalogEntryType)
```

Description

This is EntryData, a member of class EntryData.

EntryData.EntryData Constructor (int, EkEnumeration.CatalogEntryType)

C#

```
public EntryData(int languageId, EkEnumeration.CatalogEntryType entryType);
```

Visual Basic

```
Public Sub New(languageId As Integer, entryType As EkEnumeration.CatalogEntryType)
```

Description

This is EntryData, a member of class EntryData.

EntryData Properties

The properties of the EntryData class are listed here.

Public Properties

	Name	Description
	Attributes (see page 1490)	Gets or sets the attributes of the Catalog Entry.

	Comment (see page 1490)	This is Comment, a member of class EntryData.
	ContentStatus (see page 1490)	The status of the content information.
	CurrencyId (see page 1490)	Gets or sets the Id (see page 1492) of the currency used for the current Sale and List prices.
	DateModified (see page 1490)	This is DateModified, a member of class EntryData.
	Dimensions (see page 1490)	Gets or sets the dimensions of the product. Used for shipping.
	DisableInventoryManagement (see page 1491)	Gets or sets the disable inventory management flag. If true, this product does not track inventory and no inventory management is performed.
	EndDate (see page 1491)	Get or sets the enddate of the EntryData (see page 1487) object.
	EndDateAction (see page 1491)	This is EndDateAction, a member of class EntryData.
V	EntryType (see page 1491)	The type of entry.
	FolderId (see page 1492)	Gets or sets the folder ID of the EntryData (see page 1487) object.
	GoLive (see page 1492)	Get or sets the golive of the EntryData (see page 1487) object.
	Html (see page 1492)	Get or sets the html/xml of the EntryData (see page 1487) object.
	Id (see page 1492)	Gets or sets the ID of the EntryData (see page 1487) object.
	Image (see page 1493)	This is Image, a member of class EntryData.
	ImageThumbnail (see page 1493)	This is ImageThumbnail, a member of class EntryData.
	IsArchived (see page 1493)	Get or sets the Archive flag of the EntryData (see page 1487) object.
	IsBuyable (see page 1493)	Get or sets whether the entry is buyable.
	IsMarkedForDeletion (see page 1493)	Get or sets the MarkForDeletion flag on the EntryData (see page 1487) object.
	IsPublished (see page 1494)	This is IsPublished, a member of class EntryData.
	IsTangible (see page 1494)	Indicates if this product is a tangible good.
	LanguageId (see page 1494)	Gets or sets the language of the EntryData (see page 1487) object.
	LastEditorFirstName (see page 1494)	This is LastEditorFirstName, a member of class EntryData.
	LastEditorId (see page 1494)	This is LastEditorId, a member of class EntryData.
	LastEditorLastName (see page 1495)	This is LastEditorLastName, a member of class EntryData.
	ListPrice (see page 1495)	Gets the current List price from the Pricing (see page 1496) object. To set an item's price, Use the Pricing (see page 1496) property. Create a TierPrice with a quantity of 1. Use the PricingData (see page 1540)(CurrencyId (see page 1490), SalePrice (see page 1497), ListPrice) constructor.
	Media (see page 1495)	This is Media, a member of class EntryData.
	Metadata (see page 1495)	This is Metadata, a member of class EntryData.
	NumberSold (see page 1495)	Gets or sets the Number Sold for this entry.
	Pricing (see page 1496)	The entry pricing, including currencies and quantity.
	ProductType (see page 1496)	This is ProductType, a member of class EntryData.
	QuantityMultiple (see page 1496)	The has these number of items in it.
	Quicklink (see page 1496)	Gets or sets the entry's Quicklink.
	SalePrice (see page 1497)	Gets the current Sale price from the Pricing (see page 1496) object. To set an item's sale price, Use the Pricing (see page 1496) property. Create a TierPrice with a quantity of 1. Use the PricingData (see page 1540)(CurrencyId (see page 1490), SalePrice, ListPrice (see page 1495)) constructor.
	Sku (see page 1497)	Get or sets the sku of the EntryData (see page 1487) object.
	Status (see page 1497)	The status of the entry information.
	StatusLanguage (see page 1497)	The language that the entry is checked out to, if any.
	Summary (see page 1497)	Get or sets the summary of the EntryData (see page 1487) object.
	TaxClassId (see page 1498)	This is TaxClassId, a member of class EntryData.
	TemplateId (see page 1498)	This is TemplateId, a member of class EntryData.
	Title (see page 1498)	Get or sets the title of the EntryData (see page 1487) object.
	Url (see page 1498)	Get or sets the url of the EntryData (see page 1487) object.
	VersionId (see page 1498)	The version Id (see page 1492) of the catalog entry. 0 is the current version.
	Weight (see page 1499)	Gets or sets the weight of the product. Used for shipping.

EntryData.Attributes Property

Gets or sets the attributes of the Catalog Entry.

C#

```
public List<EntryAttributeData> Attributes;
```

Visual Basic

```
Public Property Attributes() As List<EntryAttributeData>
```

EntryData.Comment Property

C#

```
public string Comment;
```

Visual Basic

```
Public Property Comment() As string
```

Description

This is Comment, a member of class EntryData.

EntryData.ContentStatus Property

The status of the content information.

C#

```
public string ContentStatus;
```

Visual Basic

```
Public Property ContentStatus() As string
```

EntryData.CurrencyId Property

Gets or sets the Id (see page 1492) of the currency used for the current Sale and List prices.

C#

```
public long CurrencyId;
```

Visual Basic

```
Public Property CurrencyId() As long
```

EntryData.DateModified Property

C#

```
public DateTime DateModified;
```

Visual Basic

```
Public Property DateModified() As DateTime
```

Description

This is DateModified, a member of class EntryData.

EntryData.Dimensions Property

Gets or sets the dimensions of the product. Used for shipping.

C#

```
public Dimensions Dimensions;
```

Visual Basic

```
Public Property Dimensions() As Dimensions
```

EntryData.DisableInventoryManagement Property

Gets or sets the disable inventory management flag. If true, this product does not track inventory and no inventory management is performed.

C#

```
public bool DisableInventoryManagement;
```

Visual Basic

```
Public Property DisableInventoryManagement() As bool
```

EntryData.EndDate Property

Get or sets the enddate of the EntryData (see page 1487) object.

C#

```
public DateTime EndDate;
```

Visual Basic

```
Public Property EndDate() As DateTime
```

Returns

Datetime of the entry enddate.

EntryData.EndDateAction Property

C#

```
public int EndDateAction;
```

Visual Basic

```
Public Property EndDateAction() As Integer
```

Description

This is EndDateAction, a member of class EntryData.

EntryData.EntryType Property

The type of entry.

C#

```
public virtual EkEnumeration.CatalogEntryType EntryType;
```

Visual Basic

```
Public virtual Property EntryType() As EkEnumeration.CatalogEntryType
```

Returns

Integer of the entry quantity multiple.

EntryData.FolderId Property

Gets or sets the folder ID of the EntryData (see page 1487) object.

C#

```
public long FolderId;
```

Visual Basic

```
Public Property FolderId() As long
```

Returns

ID of the entry folder.

EntryData.GoLive Property

Get or sets the golive of the EntryData (see page 1487) object.

C#

```
public DateTime GoLive;
```

Visual Basic

```
Public Property GoLive() As DateTime
```

Returns

Datetime of the entry golive.

EntryData.Html Property

Get or sets the html/xml of the EntryData (see page 1487) object.

C#

```
public string Html;
```

Visual Basic

```
Public Property Html() As string
```

Returns

String of the entry html.

EntryData.Id Property

Gets or sets the ID of the EntryData (see page 1487) object.

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

Returns

ID of the entry title.

EntryData.Image Property

C#

```
public string Image;
```

Visual Basic

```
Public Property Image() As string
```

Description

This is Image, a member of class EntryData.

EntryData.ImageThumbnail Property

C#

```
public string ImageThumbnail;
```

Visual Basic

```
Public Property ImageThumbnail() As string
```

Description

This is ImageThumbnail, a member of class EntryData.

EntryData.IsArchived Property

Get or sets the Archive flag of the EntryData (see page 1487) object.

C#

```
public bool IsArchived;
```

Visual Basic

```
Public Property IsArchived() As bool
```

Returns

Boolean.

EntryData.IsBuyable Property

Get or sets whether the entry is buyable.

C#

```
public bool IsBuyable;
```

Visual Basic

```
Public Property IsBuyable() As bool
```

Returns

Boolean of the entry flag Buyable.

EntryData.IsMarkedForDeletion Property

Get or sets the MarkForDeletion flag on the EntryData (see page 1487) object.

C#

```
public bool IsMarkedForDeletion;
```

Visual Basic

```
Public Property IsMarkedForDeletion() As bool
```

Returns

Boolean of the entry flag MarkForDeletion.

EntryData.IsPublished Property

C#

```
public bool IsPublished;
```

Visual Basic

```
Public Property IsPublished() As bool
```

Description

This is IsPublished, a member of class EntryData.

EntryData.IsTangible Property

Indicates if this product is a tangible good.

C#

```
public bool IsTangible;
```

Visual Basic

```
Public Property IsTangible() As bool
```

Returns

Boolean value indicating if this product is a tangible good.

EntryData.LanguageId Property

Gets or sets the language of the EntryData (see page 1487) object.

C#

```
public int LanguageId;
```

Visual Basic

```
Public Property LanguageId() As Integer
```

EntryData.LastEditorFirstName Property

C#

```
public string LastEditorFirstName;
```

Visual Basic

```
Public Property LastEditorFirstName() As string
```

Description

This is LastEditorFirstName, a member of class EntryData.

EntryData.LastEditorId Property

C#

```
public long LastEditorId;
```

Visual Basic

```
Public Property LastEditorId() As long
```

Description

This is LastEditorId, a member of class EntryData.

EntryData.LastEditorLastName Property

C#

```
public string LastEditorLastName;
```

Visual Basic

```
Public Property LastEditorLastName() As string
```

Description

This is LastEditorLastName, a member of class EntryData.

EntryData.ListPrice Property

Gets the current List price from the Pricing (see page 1496) object. To set an item's price, Use the Pricing (see page 1496) property. Create a TierPrice with a quantity of 1. Use the PricingData (see page 1540)(CurrencyId (see page 1490), SalePrice (see page 1497), ListPrice) constructor.

C#

```
public decimal ListPrice;
```

Visual Basic

```
Public Property ListPrice() As decimal
```

EntryData.Media Property

C#

```
public MediaGalleryData Media;
```

Visual Basic

```
Public Property Media() As MediaGalleryData
```

Description

This is Media, a member of class EntryData.

EntryData.Metadata Property

C#

```
public List<ContentMetaData> Metadata;
```

Visual Basic

```
Public Property Metadata() As List<ContentMetaData>
```

Description

This is Metadata, a member of class EntryData.

EntryData.NumberSold Property

Gets or sets the Number Sold for this entry.

C#

```
public long NumberSold;
```

Visual Basic

```
Public Property NumberSold() As long
```

Returns

A content item's Quicklink (see page 1496) information.

EntryData.Pricing Property

The entry pricing, including currencies and quantity.

C#

```
public PricingData Pricing;
```

Visual Basic

```
Public Property Pricing() As PricingData
```

Returns

PricingData (see page 1540) of the catalog entry.

EntryData.ProductType Property

C#

```
public ProductTypeData ProductType;
```

Visual Basic

```
Public Property ProductType() As ProductTypeData
```

Description

This is ProductType, a member of class EntryData.

EntryData.QuantityMultiple Property

The has these number of items in it.

C#

```
public int QuantityMultiple;
```

Visual Basic

```
Public Property QuantityMultiple() As Integer
```

Returns

Integer of the entry quantity multiple.

EntryData.Quicklink Property

Gets or sets the entry's Quicklink.

C#

```
public string Quicklink;
```

Visual Basic

```
Public Property Quicklink() As string
```

Returns

A content item's Quicklink information.

EntryData.SalePrice Property

Gets the current Sale price from the Pricing (see page 1496) object. To set an item's sale price, Use the Pricing (see page 1496) property. Create a TierPrice with a quantity of 1. Use the PricingData (see page 1540)(CurrencyId (see page 1490), SalePrice, ListPrice (see page 1495)) constructor.

C#

```
public decimal SalePrice;
```

Visual Basic

```
Public Property SalePrice() As decimal
```

EntryData.Sku Property

Get or sets the sku of the EntryData (see page 1487) object.

C#

```
public string Sku;
```

Visual Basic

```
Public Property Sku() As string
```

Returns

String of the entry sku.

EntryData.Status Property

The status of the entry information.

C#

```
public string Status;
```

Visual Basic

```
Public Property Status() As string
```

EntryData.StatusLanguage Property

The language that the entry is checked out to, if any.

C#

```
public int StatusLanguage;
```

Visual Basic

```
Public Property StatusLanguage() As Integer
```

EntryData.Summary Property

Get or sets the summary of the EntryData (see page 1487) object.

C#

```
public string Summary;
```

Visual Basic

```
Public Property Summary() As string
```

Returns

String of the entry summary.

EntryData.TaxClassId Property

C#

```
public long TaxClassId;
```

Visual Basic

```
Public Property TaxClassId() As long
```

Description

This is TaxClassId, a member of class EntryData.

EntryData.TemplateId Property

C#

```
public long TemplateId;
```

Visual Basic

```
Public Property TemplateId() As long
```

Description

This is TemplateId, a member of class EntryData.

EntryData.Title Property

Get or sets the title of the EntryData (see page 1487) object.

C#

```
public string Title;
```

Visual Basic

```
Public Property Title() As string
```

Returns

String of the entry title.

EntryData.Url Property

Get or sets the url of the EntryData (see page 1487) object.

C#

```
public string Url;
```

Visual Basic

```
Public Property Url() As string
```

Returns

String of the entry url.

EntryData.VersionId Property

The version Id (see page 1492) of the catalog entry. 0 is the current version.

C#

```
public long VersionId;
```

Visual Basic

```
Public Property VersionId() As long
```

EntryData.Weight Property

Gets or sets the weight of the product. Used for shipping.

C#

```
public Weight Weight;
```

Visual Basic

```
Public Property Weight() As Weight
```

EntryGroupData Class

Data class representing Catalog Entry Groups in system.

Class Hierarchy

Ektron.Cms.Commerce.EntryGroupData

C#

```
public class EntryGroupData;
```

Visual Basic

```
Public Class EntryGroupData
```

File

EntryGroupData.cs

EntryGroupData Constructor

This is the overview for the EntryGroupData constructor overload.

Overload List

	Name	Description
	EntryGroupData.EntryGroupData () (see page 1499)	This is EntryGroupData, a member of class EntryGroupData.
	EntryGroupData.EntryGroupData (long, string, string, long, EkEnumeration.EntryGroupType) (see page 1500)	

EntryGroupData.EntryGroupData Constructor ()

C#

```
public EntryGroupData();
```

Visual Basic

```
Public Sub New()
```

Description

This is EntryGroupData, a member of class EntryGroupData.

EntryGroupData.EntryGroupData Constructor (long, string, string, long, EkEnumeration.EntryGroupType)

C#

```
public EntryGroupData(long id, string groupName, string description, long entryId,
EkEnumeration.EntryGroupType groupType);
```

Visual Basic

```
Public Sub New(id As long, groupName As string, description As string, entryId As long,
groupType As EkEnumeration.EntryGroupType)
```

Parameters

Parameters	Description
long id	id of entry group.
string description	description of entry group.
long entryId	Id (see page 1501) of entry owning group
groupname	name of entry group.
grouptype	type of entry group.

EntryGroupData Properties

The properties of the EntryGroupData class are listed here.

Public Properties

	Name	Description
	Description (see page 1500)	Gets or sets the description of Entry Group.
	DisplayOrder (see page 1500)	Gets or sets the display order of the entry group.
	EntryId (see page 1501)	Gets or sets the Id (see page 1501) of Catalog Entry that owns the Entry Group.
	GroupType (see page 1501)	Gets or sets the type of Entry Group.
	Id (see page 1501)	Gets or sets the Id of Entry Group.
	Name (see page 1501)	Gets or sets the Name of Entry Group.

EntryGroupData.Description Property

Gets or sets the description of Entry Group.

C#

```
public string Description;
```

Visual Basic

```
Public Property Description() As string
```

EntryGroupData.DisplayOrder Property

Gets or sets the display order of the entry group.

C#

```
public int DisplayOrder;
```

Visual Basic

```
Public Property DisplayOrder() As Integer
```

EntryGroupData.EntryId Property

Gets or sets the Id (see page 1501) of Catalog Entry that owns the Entry Group.

C#

```
public long EntryId;
```

Visual Basic

```
Public Property EntryId() As long
```

EntryGroupData.GroupType Property

Gets or sets the type of Entry Group.

C#

```
public EkEnumeration.EntryGroupType GroupType;
```

Visual Basic

```
Public Property GroupType() As EkEnumeration.EntryGroupType
```

EntryGroupData.Id Property

Gets or sets the Id of Entry Group.

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

EntryGroupData.Name Property

Gets or sets the Name of Entry Group.

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

EntryGroupItemData Class

Class Hierarchy

Ektron.Cms.Commerce.EntryGroupItemData

C#

```
public class EntryGroupItemData;
```

Visual Basic

```
Public Class EntryGroupItemData
```

File

EntryGroupItemData.cs

Description

This is class Ektron.Cms.Commerce.EntryGroupItemData.

EntryGroupItemData Constructor

This is the overview for the EntryGroupItemData constructor overload.

Overload List

	Name	Description
	EntryGroupItemData.EntryGroupItemData () (see page 1502)	This is EntryGroupItemData, a member of class EntryGroupItemData.
	EntryGroupItemData.EntryGroupItemData (long, long, long) (see page 1502)	This is EntryGroupItemData, a member of class EntryGroupItemData.

EntryGroupItemData.EntryGroupItemData Constructor ()

C#

```
public EntryGroupItemData();
```

Visual Basic

```
Public Sub New()
```

Description

This is EntryGroupItemData, a member of class EntryGroupItemData.

EntryGroupItemData.EntryGroupItemData Constructor (long, long, long)

C#

```
public EntryGroupItemData(long id, long groupId, long entryId);
```

Visual Basic

```
Public Sub New(id As long, groupId As long, entryId As long)
```

Description

This is EntryGroupItemData, a member of class EntryGroupItemData.

EntryGroupItemData Properties

The properties of the EntryGroupItemData class are listed here.

Public Properties

	Name	Description
	DisplayOrder (see page 1502)	This is DisplayOrder, a member of class EntryGroupItemData.
	EntryId (see page 1503)	This is EntryId, a member of class EntryGroupItemData.
	GroupId (see page 1503)	This is GroupId, a member of class EntryGroupItemData.
	Id (see page 1503)	This is Id, a member of class EntryGroupItemData.

EntryGroupItemData.DisplayOrder Property

C#

```
public int DisplayOrder;
```

Visual Basic

```
Public Property DisplayOrder() As Integer
```

Description

This is DisplayOrder, a member of class EntryGroupItemData.

EntryGroupItemData.EntryId Property

C#

```
public long EntryId;
```

Visual Basic

```
Public Property EntryId() As long
```

Description

This is EntryId, a member of class EntryGroupItemData.

EntryGroupItemData.GroupId Property

C#

```
public long GroupId;
```

Visual Basic

```
Public Property GroupId() As long
```

Description

This is GroupId, a member of class EntryGroupItemData.

EntryGroupItemData.Id Property

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

Description

This is Id, a member of class EntryGroupItemData.

ExchangeRateData Class

Class Hierarchy

Ektron.Cms.Commerce.ExchangeRateData

C#

```
public class ExchangeRateData;
```

Visual Basic

```
Public Class ExchangeRateData
```

File

ExchangeRateData.cs

Description

This is class Ektron.Cms.Commerce.ExchangeRateData.

ExchangeRateData.ExchangeRateData Constructor

C#

```
public ExchangeRateData();
```

Visual Basic

```
Public Sub New()
```

Description

This is ExchangeRateData, a member of class ExchangeRateData.

ExchangeRateData Methods

The methods of the ExchangeRateData class are listed here.

Public Methods

	Name	Description
	ConvertPrice (see page 1504)	Converts a pric ein the base currency to a price in the exchange currency using the rate defined.
	Validate (see page 1504)	Validates the ExchangeRate data object for saving.

ExchangeRateData.ConvertPrice Method

Converts a pric ein the base currency to a price in the exchange currency using the rate defined.

C#

```
public decimal ConvertPrice(decimal basePrice);
```

Visual Basic

```
Public Function ConvertPrice(basePrice As decimal) As decimal
```

Parameters

Parameters	Description
decimal basePrice	price to convert.

ExchangeRateData.Validate Method

Validates the ExchangeRate data object for saving.

C#

```
public ValidationResults Validate();
```

Visual Basic

```
Public Function Validate() As ValidationResults
```

Returns

returns ValidationResult object with results.

ExchangeRateData Properties

The properties of the ExchangeRateData class are listed here.

Public Properties

	Name	Description
	BaseCurrencyId (see page 1505)	Gets or sets the Currency Id for the base currency. This is the currency that is being used as the basis for the exchange rate. i.e. If you are converting from USD to AUD, USD is the base currency
	EffectiveDate (see page 1505)	Gets or sets the date and time in which this rate becomes effective.
	ExchangeCurrencyId (see page 1505)	Gets or sets the Currency Id for the exchange currency. This is the currency that the currency amount is being converted to. If you are converting from USD to AUD, AUD is the exchange currency.
	Rate (see page 1505)	Gets or sets the actual exchange rate between to currencies. This is the rate applied to base currency to convert it to exchange currency.

ExchangeRateData.BaseCurrencyId Property

Gets or sets the Currency Id for the base currency. This is the currency that is being used as the basis for the exchange rate. i.e. If you are converting from USD to AUD, USD is the base currency

C#

```
public int BaseCurrencyId;
```

Visual Basic

```
Public Property BaseCurrencyId() As Integer
```

ExchangeRateData.EffectiveDate Property

Gets or sets the date and time in which this rate becomes effective.

C#

```
public DateTime? EffectiveDate;
```

Visual Basic

```
Public Property EffectiveDate() As DateTime?
```

ExchangeRateData.ExchangeCurrencyId Property

Gets or sets the Currency Id for the exchange currency. This is the currency that the currency amount is being converted to. If you are converting from USD to AUD, AUD is the exchange currency.

C#

```
public int ExchangeCurrencyId;
```

Visual Basic

```
Public Property ExchangeCurrencyId() As Integer
```

ExchangeRateData.Rate Property

Gets or sets the actual exchange rate between to currencies. This is the rate applied to base currency to convert it to exchange currency.

C#

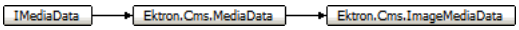
```
public decimal Rate;
```

Visual Basic

```
Public Property Rate() As decimal
```

ImageMediaData Class

Class Hierarchy



C#

```
public class ImageMediaData : MediaData;
```

Visual Basic

```
Public Class ImageMediaData
Inherits MediaData
```

File

ImageMediaData.cs

Description

This is class Ektron.Cms.ImageMediaData.

ImageMediaData Properties

The properties of the ImageMediaData class are listed here.

Public Properties

	Name	Description
	Thumbnails (see page 1506)	This is Thumbnails, a member of class ImageMediaData.

ImageMediaData.Thumbnails Property

C#

```
public List<ThumbnailData> Thumbnails;
```

Visual Basic

```
Public Property Thumbnails() As List<ThumbnailData>
```

Description

This is Thumbnails, a member of class ImageMediaData.

InventoryData Class

The InventoryData class allows you to get and set various inventory information for a catalog entry (product).

Class Hierarchy

C#

```
public class InventoryData;
```

Visual Basic

```
Public Class InventoryData
```

File

InventoryData.cs

InventoryData Properties

The properties of the InventoryData class are listed here.

Public Properties

	Name	Description
	EntryId (see page 1507)	Gets or sets the ID of the catalog entry (product) that this inventory data represents.
	ReorderLevel (see page 1507)	Gets or sets the inventory level at which stock should be reordered.
	UnitsInStock (see page 1507)	Gets or sets the the number of units currently in stock for the product.
	UnitsOnOrder (see page 1507)	Gets or sets The number of units currently on order for this product. This is the number of units that the store has ordered to replenish inventory and NOT the number of units that are currently on order by customers.

InventoryData.EntryId Property

Gets or sets the ID of the catalog entry (product) that this inventory data represents.

C#

```
public long EntryId;
```

Visual Basic

```
Public Property EntryId() As long
```

InventoryData.ReorderLevel Property

Gets or sets the inventory level at which stock should be reordered.

C#

```
public int ReorderLevel;
```

Visual Basic

```
Public Property ReorderLevel() As Integer
```

InventoryData.UnitsInStock Property

Gets or sets the the number of units currently in stock for the product.

C#

```
public int UnitsInStock;
```

Visual Basic

```
Public Property UnitsInStock() As Integer
```

InventoryData.UnitsOnOrder Property

Gets or sets The number of units currently on order for this product. This is the number of units that the store has ordered to replenish inventory and NOT the number of units that are currently on order by customers.

C#

```
public int UnitsOnOrder;
```

Visual Basic

```
Public Property UnitsOnOrder() As Integer
```

KitConfigData Class

Invitation class used for sending friend requests.

Class Hierarchy

Ektron.Cms.Commerce.KitConfigData

C#

```
public class KitConfigData;
```

Visual Basic

```
Public Class KitConfigData
```

File

KitConfigData.cs

KitConfigData.KitConfigData Constructor

C#

```
public KitConfigData();
```

Visual Basic

```
Public Sub New()
```

Description

This is KitConfigData, a member of class KitConfigData.

KitConfigData Methods

The methods of the KitConfigData class are listed here.

Public Methods

	Name	Description
	GetGroupIndex (see page 1508)	This is GetGroupIndex, a member of class KitConfigData.

KitConfigData.GetGroupIndex Method

C#

```
public int GetGroupIndex(long groupId);
```

Visual Basic

```
Public Function GetGroupIndex(groupId As long) As Integer
```

Description

This is GetGroupIndex, a member of class KitConfigData.

KitConfigData Properties

The properties of the KitConfigData class are listed here.

Public Properties

	Name	Description
	BasketItemId (see page 1509)	This is BasketItemId, a member of class KitConfigData.
	Groups (see page 1509)	This is Groups, a member of class KitConfigData.
	Id (see page 1509)	This is Id, a member of class KitConfigData.
	PriceModifier (see page 1509)	This is PriceModifier, a member of class KitConfigData.

KitConfigData.BasketItemId Property

C#

```
public long BasketItemId;
```

Visual Basic

```
Public Property BasketItemId() As long
```

Description

This is BasketItemId, a member of class KitConfigData.

KitConfigData.Groups Property

C#

```
public List<OptionGroupData> Groups;
```

Visual Basic

```
Public Property Groups() As List<OptionGroupData>
```

Description

This is Groups, a member of class KitConfigData.

KitConfigData.Id Property

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

Description

This is Id, a member of class KitConfigData.

KitConfigData.PriceModifier Property

C#

```
public decimal PriceModifier;
```

Visual Basic

```
Public Property PriceModifier() As decimal
```

Description

This is PriceModifier, a member of class KitConfigData.

KitData Class

Class Hierarchy

Ektron.Cms.Commerce.EntryData → Ektron.Cms.Commerce.KitData

C#

```
public class KitData : EntryData;
```

Visual Basic

```
Public Class KitData
Inherits EntryData
```

File

KitData.cs

KitData.KitData Constructor

C#

```
public KitData();
```

Visual Basic

```
Public Sub New()
```

Description

This is KitData, a member of class KitData.

KitData Properties

The properties of the KitData class are listed here.

Public Properties

	Name	Description
	OptionGroups (see page 1510)	This is OptionGroups, a member of class KitData.

KitData.OptionGroups Property

C#

```
public List<OptionGroupData> OptionGroups;
```

Visual Basic

```
Public Property OptionGroups() As List<OptionGroupData>
```

Description

This is OptionGroups, a member of class KitData.

MediaData Class

Class Hierarchy

IMediaData → Ektron.Cms.MediaData

C#

```
public abstract class MediaData : IMediaData;
```

Visual Basic

```
Public abstract Class MediaData
Inherits IMediaData
```

File

MediaData.cs

Description

This is class Ektron.Cms.MediaData.

MediaData Properties

The properties of the MediaData class are listed here.

Public Properties

	Name	Description
	Alt (see page 1511)	This is Alt, a member of class MediaData.
	FileName (see page 1511)	This is FileName, a member of class MediaData.
	FilePath (see page 1512)	This is FilePath, a member of class MediaData.
	Height (see page 1512)	This is Height, a member of class MediaData.
	Id (see page 1512)	This is Id, a member of class MediaData.
	MimeType (see page 1512)	This is MimeType, a member of class MediaData.
	Sequence (see page 1512)	This is Sequence, a member of class MediaData.
	Title (see page 1513)	This is Title, a member of class MediaData.
	Width (see page 1513)	This is Width, a member of class MediaData.

MediaData.Alt Property

C#

```
public string Alt;
```

Visual Basic

```
Public Property Alt() As string
```

Description

This is Alt, a member of class MediaData.

MediaData.FileName Property

C#

```
public string FileName;
```

Visual Basic

```
Public Property FileName() As string
```

Description

This is FileName, a member of class MediaData.

MediaData.FilePath Property

C#

```
public string FilePath;
```

Visual Basic

```
Public Property FilePath() As string
```

Description

This is FilePath, a member of class MediaData.

MediaData.Height Property

C#

```
public int Height;
```

Visual Basic

```
Public Property Height() As Integer
```

Description

This is Height, a member of class MediaData.

MediaData.Id Property

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

Description

This is Id, a member of class MediaData.

MediaData.MimeType Property

C#

```
public string MimeType;
```

Visual Basic

```
Public Property MimeType() As string
```

Description

This is MimeType, a member of class MediaData.

MediaData.Sequence Property

C#

```
public int Sequence;
```

Visual Basic

```
Public Property Sequence() As Integer
```

Description

This is Sequence, a member of class MediaData.

MediaData.Title Property

C#

```
public string Title;
```

Visual Basic

```
Public Property Title() As string
```

Description

This is Title, a member of class MediaData.

MediaData.Width Property

C#

```
public int Width;
```

Visual Basic

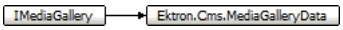
```
Public Property Width() As Integer
```

Description

This is Width, a member of class MediaData.

MediaGalleryData Class

Class Hierarchy



C#

```
public class MediaGalleryData : IMediaGallery;
```

Visual Basic

```
Public Class MediaGalleryData
Inherits IMediaGallery
```

File

MediaGalleryData.cs

Description

This is class Ektron.Cms.MediaGalleryData.

MediaGalleryData Properties

The properties of the MediaGalleryData class are listed here.

Public Properties

	Name	Description
	Images (see page 1514)	This is Images, a member of class MediaGalleryData.
	ObjectId (see page 1514)	This is ObjectId, a member of class MediaGalleryData.

MediaGalleryData.Images Property

C#

```
public List<ImageMediaData> Images;
```

Visual Basic

```
Public Property Images() As List<ImageMediaData>
```

Description

This is Images, a member of class MediaGalleryData.

MediaGalleryData.ObjectId Property

C#

```
public long ObjectId;
```

Visual Basic

```
Public Property ObjectId() As long
```

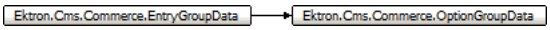
Description

This is ObjectId, a member of class MediaGalleryData.

OptionGroupData Class

Data class representing Option groups for Entry Kits.

Class Hierarchy



C#

```
public class OptionGroupData : EntryGroupData;
```

Visual Basic

```
Public Class OptionGroupData
Inherits EntryGroupData
```

File

OptionGroupData.cs

OptionGroupData Constructor

This is the overview for the OptionGroupData constructor overload.

Overload List

	Name	Description
	OptionGroupData.OptionGroupData () (see page 1515)	This is OptionGroupData, a member of class OptionGroupData.
	OptionGroupData.OptionGroupData (long, string, string, string, long) (see page 1515)	

	OptionGroupData.OptionGroupData (string, string) (see page 1515)	

OptionGroupData.OptionGroupData Constructor ()

C#

```
public OptionGroupData();
```

Visual Basic

```
Public Sub New()
```

Description

This is OptionGroupData, a member of class OptionGroupData.

OptionGroupData.OptionGroupData Constructor (long, string, string, string, long)

C#

```
public OptionGroupData(long id, string groupName, string description, string image, long
entryId);
```

Visual Basic

```
Public Sub New(id As long, groupName As string, description As string, image As string,
entryId As long)
```

Parameters

Parameters	Description
long id	Id (see page 1501) of option group.
string groupName	Name (see page 1501) of the option group.
string description	Description (see page 1500) of the OptionGroup
string image	path to image associated with option group.
long entryId	Id (see page 1501) of entry owning this option group.

OptionGroupData.OptionGroupData Constructor (string, string)

C#

```
public OptionGroupData(string groupName, string description);
```

Visual Basic

```
Public Sub New(groupName As string, description As string)
```

Parameters

Parameters	Description
string groupName	Name (see page 1501) of the option group.
string description	Description (see page 1500) of the OptionGroup
image	path to image associated with option group.

OptionGroupData Properties

The properties of the OptionGroupData class are listed here.

Public Properties

	Name	Description
	Image (see page 1516)	Gets or sets the path to the image associated with option group.

	Options (see page 1516)	gets the list of options in this option groups.

OptionGroupData.Image Property

Gets or sets the path to the image associated with option group.

C#

```
public string Image;
```

Visual Basic

```
Public Property Image() As string
```

OptionGroupData.Options Property

gets the list of options in this option groups.

C#

```
public OptionGroupItemCollection Options;
```

Visual Basic

```
Public Property Options() As OptionGroupItemCollection
```

OptionGroupItemCollection Class

Collection (see page 95) class for OptionGroupItemData (see page 1518).

Class Hierarchy

C#

```
public class OptionGroupItemCollection : Collection<OptionGroupItemData>;
```

Visual Basic

```
Public Class OptionGroupItemCollection
Inherits Collection
```

File

OptionGroupItemData.cs

OptionGroupItemCollection.OptionGroupItemCollection Constructor

C#

```
public OptionGroupItemCollection();
```

Visual Basic

```
Public Sub New()
```

Description

This is OptionGroupItemCollection, a member of class OptionGroupItemCollection.

OptionGroupItemCollection Methods

The methods of the OptionGroupItemCollection class are listed here.

Public Methods

	Name	Description
	AddRange (see page 1517)	This is AddRange, a member of class OptionGroupItemCollection.
	Find (see page 1517)	This is Find, a member of class OptionGroupItemCollection.
	FindAll (see page 1517)	This is FindAll, a member of class OptionGroupItemCollection.
	FindIndex (see page 1517)	This is FindIndex, a member of class OptionGroupItemCollection.

OptionGroupItemCollection.AddRange Method

C#

```
public void AddRange(IEnumerable<OptionGroupItemData> collection);
```

Visual Basic

```
Public Function AddRange(collection As IEnumerable<OptionGroupItemData>) As void
```

Description

This is AddRange, a member of class OptionGroupItemCollection.

OptionGroupItemCollection.Find Method

C#

```
public OptionGroupItemData Find(Predicate<OptionGroupItemData> match);
```

Visual Basic

```
Public Function Find(match As Predicate<OptionGroupItemData>) As OptionGroupItemData
```

Description

This is Find, a member of class OptionGroupItemCollection.

OptionGroupItemCollection.FindAll Method

C#

```
public List<OptionGroupItemData> FindAll(Predicate<OptionGroupItemData> match);
```

Visual Basic

```
Public Function FindAll(match As Predicate<OptionGroupItemData>) As List<OptionGroupItemData>
```

Description

This is FindAll, a member of class OptionGroupItemCollection.

OptionGroupItemCollection.FindIndex Method

C#

```
public int FindIndex(Predicate<OptionGroupItemData> match);
```

Visual Basic

```
Public Function FindIndex(match As Predicate<OptionGroupItemData>) As Integer
```

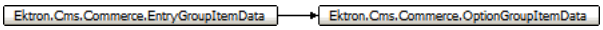
Description

This is FindIndex, a member of class OptionGroupItemCollection.

OptionGroupItemData Class

Data class representing option group items in the CMS. Option Group Items are option items for a particular Kit Grouping.

Class Hierarchy



C#

```
public class OptionGroupItemData : EntryGroupItemData;
```

Visual Basic

```
Public Class OptionGroupItemData
Inherits EntryGroupItemData
```

File

OptionGroupItemData.cs

OptionGroupItemData Constructor

This is the overview for the OptionGroupItemData constructor overload.

Overload List

	Name	Description
	OptionGroupItemData.OptionGroupItemData () (see page 1518)	This is OptionGroupItemData, a member of class OptionGroupItemData.
	OptionGroupItemData.OptionGroupItemData (long, long, string, decimal) (see page 1518)	
	OptionGroupItemData.OptionGroupItemData (string, decimal) (see page 1519)	

OptionGroupItemData.OptionGroupItemData Constructor ()

C#

```
public OptionGroupItemData();
```

Visual Basic

```
Public Sub New()
```

Description

This is OptionGroupItemData, a member of class OptionGroupItemData.

OptionGroupItemData.OptionGroupItemData Constructor (long, long, string, decimal)

C#

```
public OptionGroupItemData(long id, long groupId, string optionName, decimal
priceModification);
```

Visual Basic

```
Public Sub New(id As long, groupId As long, optionName As string, priceModification As decimal)
```

Parameters

Parameters	Description
long id	Id (see page 1503) of option group item.
long groupId	Id (see page 1503) of option group this item belongs to.
string optionName	Name (see page 1520) of the option.
decimal priceModification	Price modifier for this option.

OptionGroupItemData.OptionGroupItemData Constructor (string, decimal)

C#

```
public OptionGroupItemData(string optionName, decimal priceModification);
```

Visual Basic

```
Public Sub New(optionName As string, priceModification As decimal)
```

Parameters

Parameters	Description
string optionName	Name (see page 1520) of the option.
decimal priceModification	Price modifier for this option.

OptionGroupItemData Properties

The properties of the OptionGroupItemData class are listed here.

Public Properties

	Name	Description
	CurrencyId (see page 1519)	Gets or sets the currency Id (see page 1503).
	ExtraText (see page 1519)	Gets or sets any extra test associated with this option item.
	Name (see page 1520)	Gets or sets the name of the option item.
	PriceModification (see page 1520)	Gets or sets the price modification value for this option. i.e. this option increases the price $10.00.
	PriceModificationType (see page 1520)	Gets or sets the proce modification type for this option. i.e. is the price modification a percentage or fixed amount.

OptionGroupItemData.CurrencyId Property

Gets or sets the currency Id (see page 1503).

C#

```
public int CurrencyId;
```

Visual Basic

```
Public Property CurrencyId() As Integer
```

OptionGroupItemData.ExtraText Property

Gets or sets any extra test associated with this option item.

C#

```
public string ExtraText;
```

Visual Basic

```
Public Property ExtraText() As string
```

OptionGroupItemData.Name Property

Gets or sets the name of the option item.

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

OptionGroupItemData.PriceModification Property

Gets or sets the price modification value for this option. i.e. this option increases the price $10.00.

C#

```
public decimal PriceModification;
```

Visual Basic

```
Public Property PriceModification() As decimal
```

OptionGroupItemData.PriceModificationType Property

Gets or sets the proce modification type for this option. i.e. is the price modification a percentage or fixed amount.

C#

```
public EkEnumeration.PriceModificationType PriceModificationType;
```

Visual Basic

```
Public Property PriceModificationType() As EkEnumeration.PriceModificationType
```

OrderData Class

Class Hierarchy

Ektron.Cms.Commerce.OrderData

C#

```
public class OrderData;
```

Visual Basic

```
Public Class OrderData
```

File

OrderData.cs

OrderData Properties

The properties of the OrderData class are listed here.

Public Properties

	Name	Description
	BillingAddressId (see page 1521)	This is BillingAddressId, a member of class OrderData.
	ClientIP (see page 1521)	This is ClientIP, a member of class OrderData.
	CouponTotal (see page 1521)	Gets or sets the total coupon discount for the order.

	CouponUsed (see page 1522)	This is CouponUsed, a member of class OrderData.
	Currency (see page 1522)	This is Currency, a member of class OrderData.
	Customer (see page 1522)	This is Customer, a member of class OrderData.
	CustomerType (see page 1522)	This is CustomerType, a member of class OrderData.
	DateCompleted (see page 1522)	This is DateCompleted, a member of class OrderData.
	DateCreated (see page 1523)	This is DateCreated, a member of class OrderData.
	DateRequired (see page 1523)	This is DateRequired, a member of class OrderData.
	HandlingTotal (see page 1523)	Gets or sets the total handling charges for the order.
	HttpHost (see page 1523)	This is HttpHost, a member of class OrderData.
	Id (see page 1523)	This is Id, a member of class OrderData.
	OrderTotal (see page 1524)	Gets or sets the Order Total.
	Parts (see page 1524)	This is Parts, a member of class OrderData.
	ShippingTotal (see page 1524)	Gets or sets the total shipping charged for order.
	SpecialInstructions (see page 1524)	This is SpecialInstructions, a member of class OrderData.
	StageId (see page 1524)	This is StageId, a member of class OrderData.
	StageName (see page 1525)	This is StageName, a member of class OrderData.
	Status (see page 1525)	This is Status, a member of class OrderData.
	Subtotal (see page 1525)	Gets or sets the Order SubTotal.
	TaxTotal (see page 1525)	Gets or sets the total tax amount for the order.
	VisitorId (see page 1525)	This is VisitorId, a member of class OrderData.

OrderData.BillingAddressId Property

C#

```
public long BillingAddressId;
```

Visual Basic

```
Public Property BillingAddressId() As long
```

Description

This is BillingAddressId, a member of class OrderData.

OrderData.ClientIP Property

C#

```
public string ClientIP;
```

Visual Basic

```
Public Property ClientIP() As string
```

Description

This is ClientIP, a member of class OrderData.

OrderData.CouponTotal Property

Gets or sets the total coupon discount for the order.

C#

```
public decimal CouponTotal;
```

Visual Basic

```
Public Property CouponTotal() As decimal
```

OrderData.CouponUsed Property

C#

```
public bool CouponUsed;
```

Visual Basic

```
Public Property CouponUsed() As bool
```

Description

This is CouponUsed, a member of class OrderData.

OrderData.Currency Property

C#

```
public CurrencyData Currency;
```

Visual Basic

```
Public Property Currency() As CurrencyData
```

Description

This is Currency, a member of class OrderData.

OrderData.Customer Property

C#

```
public CustomerData Customer;
```

Visual Basic

```
Public Property Customer() As CustomerData
```

Description

This is Customer, a member of class OrderData.

OrderData.CustomerType Property

C#

```
public EkEnumeration.CustomerType CustomerType;
```

Visual Basic

```
Public Property CustomerType() As EkEnumeration.CustomerType
```

Description

This is CustomerType, a member of class OrderData.

OrderData.DateCompleted Property

C#

```
public DateTime DateCompleted;
```

Visual Basic

```
Public Property DateCompleted() As DateTime
```

Description

This is DateCompleted, a member of class OrderData.

OrderData.DateCreated Property

C#

```
public DateTime DateCreated;
```

Visual Basic

```
Public Property DateCreated() As DateTime
```

Description

This is DateCreated, a member of class OrderData.

OrderData.DateRequired Property

C#

```
public DateTime DateRequired;
```

Visual Basic

```
Public Property DateRequired() As DateTime
```

Description

This is DateRequired, a member of class OrderData.

OrderData.HandlingTotal Property

Gets or sets the total handling charges for the order.

C#

```
public decimal HandlingTotal;
```

Visual Basic

```
Public Property HandlingTotal() As decimal
```

OrderData.HttpHost Property

C#

```
public string HttpHost;
```

Visual Basic

```
Public Property HttpHost() As string
```

Description

This is HttpHost, a member of class OrderData.

OrderData.Id Property

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

Description

This is Id, a member of class OrderData.

OrderData.OrderTotal Property

Gets or sets the Order Total.

C#

```
public decimal OrderTotal;
```

Visual Basic

```
Public Property OrderTotal() As decimal
```

OrderData.Parts Property

C#

```
public List<OrderPartData> Parts;
```

Visual Basic

```
Public Property Parts() As List<OrderPartData>
```

Description

This is Parts, a member of class OrderData.

OrderData.ShippingTotal Property

Gets or sets the total shipping charged for order.

C#

```
public decimal ShippingTotal;
```

Visual Basic

```
Public Property ShippingTotal() As decimal
```

OrderData.SpecialInstructions Property

C#

```
public string SpecialInstructions;
```

Visual Basic

```
Public Property SpecialInstructions() As string
```

Description

This is SpecialInstructions, a member of class OrderData.

OrderData.StageId Property

C#

```
public int StageId;
```

Visual Basic

```
Public Property StageId() As Integer
```

Description

This is StageId, a member of class OrderData.

OrderData.StageName Property

C#

```
public string StageName;
```

Visual Basic

```
Public Property StageName() As string
```

Description

This is StageName, a member of class OrderData.

OrderData.Status Property

C#

```
public EkEnumeration.OrderStatus Status;
```

Visual Basic

```
Public Property Status() As EkEnumeration.OrderStatus
```

Description

This is Status, a member of class OrderData.

OrderData.Subtotal Property

Gets or sets the Order SubTotal.

C#

```
public decimal Subtotal;
```

Visual Basic

```
Public Property Subtotal() As decimal
```

OrderData.TaxTotal Property

Gets or sets the total tax amount for the order.

C#

```
public decimal TaxTotal;
```

Visual Basic

```
Public Property TaxTotal() As decimal
```

OrderData.VisitorId Property

C#

```
public string VisitorId;
```

Visual Basic

```
Public Property VisitorId() As string
```

Description

This is VisitorId, a member of class OrderData.

OrderLineData Class

Class Hierarchy

Ektron.Cms.Commerce.OrderLineData

C#

```
public class OrderLineData;
```

Visual Basic

```
Public Class OrderLineData
```

File

OrderLineData.cs

OrderLineData Properties

The properties of the OrderLineData class are listed here.

Public Properties

	Name	Description
	Configuration (see page 1526)	This is Configuration, a member of class OrderLineData.
	ItemType (see page 1526)	This is ItemType, a member of class OrderLineData.
	LineItemId (see page 1527)	This is LineItemId, a member of class OrderLineData.
	OrderPartId (see page 1527)	This is OrderPartId, a member of class OrderLineData.
	PriceEach (see page 1527)	Gets or sets the unit sale price for each line item.
	PriceSubTotal (see page 1527)	Gets or sets the line item subtotal. This does not reflect and discounts or adjustments.
	PriceTotal (see page 1527)	Gets or sets the total line item price. This reflects quantity and any applicable discounts.
	ProductId (see page 1528)	This is ProductId, a member of class OrderLineData.
	ProductLanguage (see page 1528)	This is ProductLanguage, a member of class OrderLineData.
	Quantity (see page 1528)	This is Quantity, a member of class OrderLineData.

OrderLineData.Configuration Property

C#

```
public OrderKitConfigData Configuration;
```

Visual Basic

```
Public Property Configuration() As OrderKitConfigData
```

Description

This is Configuration, a member of class OrderLineData.

OrderLineData.ItemType Property

C#

```
public EkEnumeration.CatalogEntryType ItemType;
```

Visual Basic

```
Public Property ItemType() As EkEnumeration.CatalogEntryType
```

Description

This is ItemType, a member of class OrderLineData.

OrderLineData.LineItemId Property

C#

```
public long LineItemId;
```

Visual Basic

```
Public Property LineItemId() As long
```

Description

This is LineItemId, a member of class OrderLineData.

OrderLineData.OrderPartId Property

C#

```
public long OrderPartId;
```

Visual Basic

```
Public Property OrderPartId() As long
```

Description

This is OrderPartId, a member of class OrderLineData.

OrderLineData.PriceEach Property

Gets or sets the unit sale price for each line item.

C#

```
public decimal PriceEach;
```

Visual Basic

```
Public Property PriceEach() As decimal
```

OrderLineData.PriceSubTotal Property

Gets or sets the line item subtotal. This does not reflect and discounts or adjustments.

C#

```
public decimal PriceSubTotal;
```

Visual Basic

```
Public Property PriceSubTotal() As decimal
```

OrderLineData.PriceTotal Property

Gets or sets the total line item price. This reflects quantity and any applicable discounts.

C#

```
public decimal PriceTotal;
```

Visual Basic

```
Public Property PriceTotal() As decimal
```

OrderLineData.ProductId Property

C#

```
public long ProductId;
```

Visual Basic

```
Public Property ProductId() As long
```

Description

This is ProductId, a member of class OrderLineData.

OrderLineData.ProductLanguage Property

C#

```
public long ProductLanguage;
```

Visual Basic

```
Public Property ProductLanguage() As long
```

Description

This is ProductLanguage, a member of class OrderLineData.

OrderLineData.Quantity Property

C#

```
public int Quantity;
```

Visual Basic

```
Public Property Quantity() As Integer
```

Description

This is Quantity, a member of class OrderLineData.

OrderPartData Class

data class representing order part sin the cms.

Class Hierarchy

Ektron.Cms.Commerce.OrderPartData

C#

```
public class OrderPartData;
```

Visual Basic

```
Public Class OrderPartData
```

File

OrderPartData.cs

OrderPartData Properties

The properties of the OrderPartData class are listed here.

Public Properties

	Name	Description
	DateShipped (see page 1529)	This is DateShipped, a member of class OrderPartData.
	HandlingTotal (see page 1529)	This is HandlingTotal, a member of class OrderPartData.
	Lines (see page 1529)	This is Lines, a member of class OrderPartData.
	OrderId (see page 1530)	This is OrderId, a member of class OrderPartData.
	PartId (see page 1530)	This is PartId, a member of class OrderPartData.
	PartTotal (see page 1530)	This is PartTotal, a member of class OrderPartData.
	ShippingAddressId (see page 1530)	This is ShippingAddressId, a member of class OrderPartData.
	ShippingTotal (see page 1530)	This is ShippingTotal, a member of class OrderPartData.
	Status (see page 1531)	This is Status, a member of class OrderPartData.
	Subtotal (see page 1531)	This is Subtotal, a member of class OrderPartData.
	SupplierId (see page 1531)	This is SupplierId, a member of class OrderPartData.
	TaxTotal (see page 1531)	This is TaxTotal, a member of class OrderPartData.
	TrackingNumber (see page 1531)	This is TrackingNumber, a member of class OrderPartData.

OrderPartData.DateShipped Property

C#

```
public DateTime DateShipped;
```

Visual Basic

```
Public Property DateShipped() As DateTime
```

Description

This is DateShipped, a member of class OrderPartData.

OrderPartData.HandlingTotal Property

C#

```
public decimal HandlingTotal;
```

Visual Basic

```
Public Property HandlingTotal() As decimal
```

Description

This is HandlingTotal, a member of class OrderPartData.

OrderPartData.Lines Property

C#

```
public List<OrderLineData> Lines;
```

Visual Basic

```
Public Property Lines() As List<OrderLineData>
```

Description

This is Lines, a member of class OrderPartData.

OrderPartData.OrderId Property

C#

```
public long OrderId;
```

Visual Basic

```
Public Property OrderId() As long
```

Description

This is OrderId, a member of class OrderPartData.

OrderPartData.PartId Property

C#

```
public long PartId;
```

Visual Basic

```
Public Property PartId() As long
```

Description

This is PartId, a member of class OrderPartData.

OrderPartData.PartTotal Property

C#

```
public decimal PartTotal;
```

Visual Basic

```
Public Property PartTotal() As decimal
```

Description

This is PartTotal, a member of class OrderPartData.

OrderPartData.ShippingAddressId Property

C#

```
public long ShippingAddressId;
```

Visual Basic

```
Public Property ShippingAddressId() As long
```

Description

This is ShippingAddressId, a member of class OrderPartData.

OrderPartData.ShippingTotal Property

C#

```
public decimal ShippingTotal;
```

Visual Basic

```
Public Property ShippingTotal() As decimal
```

Description

This is ShippingTotal, a member of class OrderPartData.

OrderPartData.Status Property

C#

```
public int Status;
```

Visual Basic

```
Public Property Status() As Integer
```

Description

This is Status, a member of class OrderPartData.

OrderPartData.Subtotal Property

C#

```
public decimal Subtotal;
```

Visual Basic

```
Public Property Subtotal() As decimal
```

Description

This is Subtotal, a member of class OrderPartData.

OrderPartData.SupplierId Property

C#

```
public long SupplierId;
```

Visual Basic

```
Public Property SupplierId() As long
```

Description

This is SupplierId, a member of class OrderPartData.

OrderPartData.TaxTotal Property

C#

```
public decimal TaxTotal;
```

Visual Basic

```
Public Property TaxTotal() As decimal
```

Description

This is TaxTotal, a member of class OrderPartData.

OrderPartData.TrackingNumber Property

C#

```
public string TrackingNumber;
```

Visual Basic

```
Public Property TrackingNumber() As string
```

Description

This is TrackingNumber, a member of class OrderPartData.

PackageData Class

Data class used for representing package sizes used for shipping items.

Class Hierarchy

Ektron.Cms.Commerce.PackageData

C#

```
public class PackageData;
```

Visual Basic

```
Public Class PackageData
```

File

PackageData.cs

PackageData Constructor

This is the overview for the PackageData constructor overload.

Overload List

	Name	Description
	PackageData.PackageData () (see page 1532)	This is PackageData, a member of class PackageData.
	PackageData.PackageData (string, Dimensions, Weight) (see page 1532)	
	PackageData.PackageData (string, Dimensions, Weight, long) (see page 1533)	

PackageData.PackageData Constructor ()

C#

```
public PackageData();
```

Visual Basic

```
Public Sub New()
```

Description

This is PackageData, a member of class PackageData.

PackageData.PackageData Constructor (string, Dimensions, Weight)

C#

```
public PackageData(string name, Dimensions dimensions, Weight maxWeight);
```

Visual Basic

```
Public Sub New(name As string, dimensions As Dimensions, maxWeight As Weight)
```

Parameters

Parameters	Description
string name	the name of the package
Dimensions dimensions	dimensions of the package
Weight maxWeight	maximum weight capacity of the package

PackageData.PackageData Constructor (string, Dimensions, Weight, long)

C#

```
public PackageData(string name, Dimensions dimensions, Weight maxWeight, long id);
```

Visual Basic

```
Public Sub New(name As string, dimensions As Dimensions, maxWeight As Weight, id As long)
```

Parameters

Parameters	Description
string name	the name of the package
Dimensions dimensions	dimensions of the package
Weight maxWeight	maximum weight capacity of the package
long id	the Id (see page 1533) of the package.

PackageData Properties

The properties of the PackageData class are listed here.

Public Properties

	Name	Description
	Dimensions (see page 1533)	gets or sets the dimensions of the package.
	Id (see page 1533)	Gets or sets the Id of the package.
	MaxWeight (see page 1533)	gets or sets the maximum weight capacity of the package.
	Name (see page 1534)	Gets or sets the name of the package.

PackageData.Dimensions Property

gets or sets the dimensions of the package.

C#

```
public Dimensions Dimensions;
```

Visual Basic

```
Public Property Dimensions() As Dimensions
```

PackageData.Id Property

Gets or sets the Id of the package.

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

PackageData.MaxWeight Property

gets or sets the maximum weight capacity of the package.

C#

```
public Weight MaxWeight;
```

Visual Basic

```
Public Property MaxWeight() As Weight
```

PackageData.Name Property

Gets or sets the name of the package.

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

PaymentAuthorization Class

Data class representing payment authorization from gateway.

Class Hierarchy

Ektron.Cms.Commerce.PaymentAuthorization

C#

```
public class PaymentAuthorization;
```

Visual Basic

```
Public Class PaymentAuthorization
```

File

PaymentAuthorization.cs

PaymentAuthorization Properties

The properties of the PaymentAuthorization class are listed here.

Public Properties

	Name	Description
	AuthorizedOn (see page 1534)	Gets or sets the authorization date.
	CapturedOn (see page 1535)	Gets or sets the Capture Date.
	CreditedOn (see page 1535)	Gets or sets the Credited Date.
	Gateway (see page 1535)	Gets or sets the name of the gateway used for this authorization.
	Message (see page 1535)	Gets or sets a success or error message for the authorization.
	RecurringTransactionId (see page 1535)	Gets or sets the Transaction id for a recurring transaction.
	TransactionId (see page 1535)	Gets or sets the Transaction Id for this authorization.
	VoidedOn (see page 1536)	Gets or sets the Voided Date.

PaymentAuthorization.AuthorizedOn Property

Gets or sets the authorization date.

C#

```
public DateTime AuthorizedOn;
```

Visual Basic

```
Public Property AuthorizedOn() As DateTime
```

PaymentAuthorization.CapturedOn Property

Gets or sets the Capture Date.

C#

```
public DateTime CapturedOn;
```

Visual Basic

```
Public Property CapturedOn() As DateTime
```

PaymentAuthorization.CreditedOn Property

Gets or sets the Credited Date.

C#

```
public DateTime CreditedOn;
```

Visual Basic

```
Public Property CreditedOn() As DateTime
```

PaymentAuthorization.Gateway Property

Gets or sets the name of the gateway used for this authorization.

C#

```
public string Gateway;
```

Visual Basic

```
Public Property Gateway() As string
```

PaymentAuthorization.Message Property

Gets or sets a success or error message for the authorization.

C#

```
public string Message;
```

Visual Basic

```
Public Property Message() As string
```

PaymentAuthorization.RecurringTransactionId Property

Gets or sets the Transaction id for a recurring transaction.

C#

```
public string RecurringTransactionId;
```

Visual Basic

```
Public Property RecurringTransactionId() As string
```

PaymentAuthorization.TransactionId Property

Gets or sets the Transaction Id for this authorization.

C#

```
public string TransactionId;
```

Visual Basic

```
Public Property TransactionId() As string
```

PaymentAuthorization.VoidedOn Property

Gets or sets the Voided Date.

C#

```
public DateTime VoidedOn;
```

Visual Basic

```
Public Property VoidedOn() As DateTime
```

PaymentMethod Class

Base class used for representing payment methods for use by payment provider.

Class Hierarchy

Ektron.Cms.Commerce.PaymentMethod

C#

```
public class PaymentMethod;
```

Visual Basic

```
Public Class PaymentMethod
```

File

PaymentMethod.cs

PaymentMethod Methods

The methods of the PaymentMethod class are listed here.

Public Methods

	Name	Description
	GetPayment (see page 1536)	Returns a PaymentMethod (see page 1536) based upon NameValueCollection. This would be used by the checkout control to get the paymentMethod posted back from the page.

PaymentMethod.GetPayment Method

Returns a PaymentMethod (see page 1536) based upon NameValueCollection. This would be used by the checkout control to get the paymentMethod posted back from the page.

C#

```
public virtual PaymentMethod GetPayment(System.Collections.Specialized.NameValueCollection
postBackData);
```

Visual Basic

```
Public virtual Function GetPayment(postBackData As
System.Collections.Specialized.NameValueCollection) As PaymentMethod
```

Parameters

Parameters	Description
System.Collections.Specialized.NameValueCollection postBackData	

PaymentMethod Properties

The properties of the PaymentMethod class are listed here.

Public Properties

	Name	Description
	BillingAddress (see page 1537)	Gets or sets the Billing Address of the party being authorized.
	BillingFirstName (see page 1537)	Gets or sets the first name of the billable party.
	BillingLastName (see page 1537)	Gets or sets the last name of the billable party.

PaymentMethod.BillingAddress Property

Gets or sets the Billing Address of the party being authorized.

C#

```
public virtual AddressData BillingAddress;
```

Visual Basic

```
Public virtual Property BillingAddress() As AddressData
```

PaymentMethod.BillingFirstName Property

Gets or sets the first name of the billable party.

C#

```
public virtual string BillingFirstName;
```

Visual Basic

```
Public virtual Property BillingFirstName() As string
```

PaymentMethod.BillingLastName Property

Gets or sets the last name of the billable party.

C#

```
public virtual string BillingLastName;
```

Visual Basic

```
Public virtual Property BillingLastName() As string
```

PipelineStageData Class

Class Hierarchy

Ektron.Cms.Commerce.PipelineStageData

C#

```
public class PipelineStageData;
```

Visual Basic

```
Public Class PipelineStageData
```

File

PipelineStageData.cs

PipelineStageData Properties

The properties of the PipelineStageData class are listed here.

Public Properties

	Name	Description
	Name (see page 1538)	This is Name, a member of class PipelineStageData.
	Sequence (see page 1538)	This is Sequence, a member of class PipelineStageData.
	StatusId (see page 1538)	This is StatusId, a member of class PipelineStageData.

PipelineStageData.Name Property

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

Description

This is Name, a member of class PipelineStageData.

PipelineStageData.Sequence Property

C#

```
public int Sequence;
```

Visual Basic

```
Public Property Sequence() As Integer
```

Description

This is Sequence, a member of class PipelineStageData.

PipelineStageData.StatusId Property

C#

```
public long StatusId;
```

Visual Basic

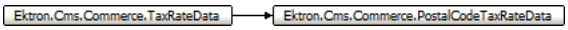
```
Public Property StatusId() As long
```

Description

This is StatusId, a member of class PipelineStageData.

PostalCodeTaxRateData Class

Class Hierarchy



C#

```
public class PostalCodeTaxRateData : TaxRateData;
```

Visual Basic

```
Public Class PostalCodeTaxRateData
Inherits TaxRateData
```

File

PostalCodeTaxRateData.cs

Description

This is class Ektron.Cms.Commerce.PostalCodeTaxRateData.

PostalCodeTaxRateData Constructor

This is the overview for the PostalCodeTaxRateData constructor overload.

Overload List

	Name	Description
	PostalCodeTaxRateData.PostalCodeTaxRateData () (see page 1539)	This is PostalCodeTaxRateData, a member of class PostalCodeTaxRateData.
	PostalCodeTaxRateData.PostalCodeTaxRateData (string, long, long, decimal) (see page 1539)	This is PostalCodeTaxRateData, a member of class PostalCodeTaxRateData.
	PostalCodeTaxRateData.PostalCodeTaxRateData (string, long, long, decimal, long) (see page 1540)	This is PostalCodeTaxRateData, a member of class PostalCodeTaxRateData.

PostalCodeTaxRateData.PostalCodeTaxRateData Constructor ()

C#

```
public PostalCodeTaxRateData();
```

Visual Basic

```
Public Sub New()
```

Description

This is PostalCodeTaxRateData, a member of class PostalCodeTaxRateData.

PostalCodeTaxRateData.PostalCodeTaxRateData Constructor (string, long, long, decimal)

C#

```
public PostalCodeTaxRateData(string postalCode, long regionId, long taxClassId, decimal rate);
```

Visual Basic

```
Public Sub New(postalCode As string, regionId As long, taxClassId As long, rate As decimal)
```

Description

This is PostalCodeTaxRateData, a member of class PostalCodeTaxRateData.

PostalCodeTaxRateData.PostalCodeTaxRateData Constructor (string, long, long, decimal, long)

C#

```
public PostalCodeTaxRateData(string postalCode, long regionId, long taxClassId, decimal rate,
long id);
```

Visual Basic

```
Public Sub New(postalCode As string, regionId As long, taxClassId As long, rate As decimal, id
As long)
```

Description

This is PostalCodeTaxRateData, a member of class PostalCodeTaxRateData.

PostalCodeTaxRateData Properties

The properties of the PostalCodeTaxRateData class are listed here.

Public Properties

	Name	Description
	PostalCode (see page 1540)	Gets or sets the postal code associated with this tax rate.
	RegionId (see page 1540)	Gets or sets the regionid asociated with this postal code.

PostalCodeTaxRateData.PostalCode Property

Gets or sets the postal code associated with this tax rate.

C#

```
public string PostalCode;
```

Visual Basic

```
Public Property PostalCode() As string
```

PostalCodeTaxRateData.RegionId Property

Gets or sets the regionid asociated with this postal code.

C#

```
public long RegionId;
```

Visual Basic

```
Public Property RegionId() As long
```

PricingData Class

Class Hierarchy

Ektron.Cms.Commerce.PricingData

C#

```
public class PricingData;
```

Visual Basic

```
Public Class PricingData
```

File

PricingData.cs

Description

This is class Ektron.Cms.Commerce.PricingData.

PricingData Constructor

This is the overview for the PricingData constructor overload.

Overload List

	Name	Description
	PricingData.PricingData () (see page 1541)	This is PricingData, a member of class PricingData.
	PricingData.PricingData (List<CurrencyPricingData>) (see page 1541)	This is PricingData, a member of class PricingData.
	PricingData.PricingData (long, decimal, decimal) (see page 1541)	Creates a PricingData with a single CurrencyPricingData (see page 1474) that contains a single tierprice with the supplied sale price.

PricingData.PricingData Constructor ()

C#

```
public PricingData();
```

Visual Basic

```
Public Sub New()
```

Description

This is PricingData, a member of class PricingData.

PricingData.PricingData Constructor (List<CurrencyPricingData>)

C#

```
public PricingData(List<CurrencyPricingData> currencyPricelist);
```

Visual Basic

```
Public Sub New(currencyPricelist As List<CurrencyPricingData>)
```

Description

This is PricingData, a member of class PricingData.

PricingData.PricingData Constructor (long, decimal, decimal)

Creates a PricingData with a single CurrencyPricingData (see page 1474) that contains a single tierprice with the supplied sale price.

C#

```
public PricingData(long currencyId, decimal salePrice, decimal listPrice);
```

Visual Basic

```
Public Sub New(currencyId As long, salePrice As decimal, listPrice As decimal)
```

Parameters

Parameters	Description
long currencyId	Id of currency to create pricing for.
decimal salePrice	sale Price for single tier item.
decimal listPrice	List Price for currency.

PricingData Methods

The methods of the PricingData class are listed here.

Public Methods

	Name	Description
	GetCurrencyById (see page 1542)	This is GetCurrencyById, a member of class PricingData.
	GetCurrencyByIso (see page 1542)	This is GetCurrencyByIso, a member of class PricingData.
	GetCurrencyByIsoNumber (see page 1543)	This is GetCurrencyByIsoNumber, a member of class PricingData.
	GetListPrice (see page 1543)	This is GetListPrice, a member of class PricingData.
	GetListPriceByCurrencyId (see page 1543)	This is GetListPriceByCurrencyId, a member of class PricingData.
	GetListPriceByIsoNumber (see page 1543)	This is GetListPriceByIsoNumber, a member of class PricingData.
	GetSalePrice (see page 1543)	This is GetSalePrice, a member of class PricingData.
	GetSalePriceByCurrencyId (see page 1544)	This is GetSalePriceByCurrencyId, a member of class PricingData.
	GetSalePriceByIsoNumber (see page 1544)	This is GetSalePriceByIsoNumber, a member of class PricingData.
	Validate (see page 1544)	This is Validate, a member of class PricingData.

PricingData.GetCurrencyById Method

C#

```
public CurrencyPricingData GetCurrencyById(long currencyId);
```

Visual Basic

```
Public Function GetCurrencyById(currencyId As long) As CurrencyPricingData
```

Description

This is GetCurrencyById, a member of class PricingData.

PricingData.GetCurrencyByIso Method

C#

```
public CurrencyPricingData GetCurrencyByIso(string alphaIsoCode);
```

Visual Basic

```
Public Function GetCurrencyByIso(alphaIsoCode As string) As CurrencyPricingData
```

Description

This is GetCurrencyByIso, a member of class PricingData.

PricingData.GetCurrencyByIsoNumber Method

C#

```
public CurrencyPricingData GetCurrencyByIsoNumber(int numericIsoCode);
```

Visual Basic

```
Public Function GetCurrencyByIsoNumber(numericIsoCode As Integer) As CurrencyPricingData
```

Description

This is GetCurrencyByIsoNumber, a member of class PricingData.

PricingData.GetListPrice Method

C#

```
public decimal GetListPrice(string alphaIsoCode);
```

Visual Basic

```
Public Function GetListPrice(alphaIsoCode As string) As decimal
```

Description

This is GetListPrice, a member of class PricingData.

PricingData.GetListPriceByCurrencyId Method

C#

```
public decimal GetListPriceByCurrencyId(long currencyId);
```

Visual Basic

```
Public Function GetListPriceByCurrencyId(currencyId As long) As decimal
```

Description

This is GetListPriceByCurrencyId, a member of class PricingData.

PricingData.GetListPriceByIsoNumber Method

C#

```
public decimal GetListPriceByIsoNumber(int numericIsoCode);
```

Visual Basic

```
Public Function GetListPriceByIsoNumber(numericIsoCode As Integer) As decimal
```

Description

This is GetListPriceByIsoNumber, a member of class PricingData.

PricingData.GetSalePrice Method

C#

```
public decimal GetSalePrice(int quantity, string alphaIsoCode);
```

Visual Basic

```
Public Function GetSalePrice(quantity As Integer, alphaIsoCode As string) As decimal
```

Description

This is GetSalePrice, a member of class PricingData.

PricingData.GetSalePriceByCurrencyId Method

C#

```
public decimal GetSalePriceByCurrencyId(int quantity, long currencyId);
```

Visual Basic

```
Public Function GetSalePriceByCurrencyId(quantity As Integer, currencyId As long) As decimal
```

Description

This is GetSalePriceByCurrencyId, a member of class PricingData.

PricingData.GetSalePriceByIsoNumber Method

C#

```
public decimal GetSalePriceByIsoNumber(int quantity, int numericIsoCode);
```

Visual Basic

```
Public Function GetSalePriceByIsoNumber(quantity As Integer, numericIsoCode As Integer) As
decimal
```

Description

This is GetSalePriceByIsoNumber, a member of class PricingData.

PricingData.Validate Method

C#

```
public ValidationResults Validate();
```

Visual Basic

```
Public Function Validate() As ValidationResults
```

Description

This is Validate, a member of class PricingData.

PricingData Properties

The properties of the PricingData class are listed here.

Public Properties

	Name	Description
	CurrencyPricelist (see page 1545)	Gets list of CurrencyPriceData objects associated with this PricingData (see page 1540) object.
	IsRecurringPrice (see page 1545)	Gets the Is Recurring Price flag. If true, this price is for recurring billing.
	Recurrence (see page 1545)	Gets or sets the recurrence pattern for this price if it supports recurring billing.

PricingData.CurrencyPricelist Property

Gets list of CurrencyPriceData objects associated with this PricingData (see page 1540) object.

C#

```
public List<Ektron.Cms.Commerce.CurrencyPricingData> CurrencyPricelist;
```

Visual Basic

```
Public Property CurrencyPricelist() As List<Ektron.Cms.Commerce.CurrencyPricingData>
```

PricingData.IsRecurringPrice Property

Gets the Is Recurring Price flag. If true, this price is for recurring billing.

C#

```
public bool IsRecurringPrice;
```

Visual Basic

```
Public Property IsRecurringPrice() As bool
```

PricingData.Recurrence Property

Gets or sets the recurrence pattern for this price if it supports recurring billing.

C#

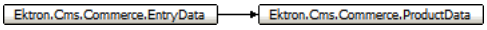
```
public RecurrenceData Recurrence;
```

Visual Basic

```
Public Property Recurrence() As RecurrenceData
```

ProductData Class

Class Hierarchy



C#

```
public class ProductData : EntryData;
```

Visual Basic

```
Public Class ProductData
Inherits EntryData
```

File

ProductData.cs

ProductData Constructor

This is the overview for the ProductData constructor overload.

Overload List

	Name	Description
	ProductData.ProductData () (see page 1546)	This is ProductData, a member of class ProductData.

	ProductData.ProductData (int) (see page 1546)	This is ProductData, a member of class ProductData.

ProductData.ProductData Constructor ()

C#

```
public ProductData();
```

Visual Basic

```
Public Sub New()
```

Description

This is ProductData, a member of class ProductData.

ProductData.ProductData Constructor (int)

C#

```
public ProductData(int languageId);
```

Visual Basic

```
Public Sub New(languageId As Integer)
```

Description

This is ProductData, a member of class ProductData.

ProductData Properties

The properties of the ProductData class are listed here.

Public Properties

	Name	Description
	Variants (see page 1546)	This is Variants, a member of class ProductData.

ProductData.Variants Property

C#

```
public List<ProductVariantData> Variants;
```

Visual Basic

```
Public Property Variants() As List<ProductVariantData>
```

Description

This is Variants, a member of class ProductData.

ProductTypeAttributeData Class

Class Hierarchy

Ektron.Cms.Commerce.ProductTypeAttributeData

C#

```
public class ProductTypeAttributeData;
```

Visual Basic

```
Public Class ProductTypeAttributeData
```

File

ProductAttributeData.cs

Description

This is class Ektron.Cms.Commerce.ProductTypeAttributeData.

ProductTypeAttributeData Constructor

This is the overview for the ProductTypeAttributeData constructor overload.

Overload List

	Name	Description
	ProductTypeAttributeData.ProductTypeAttributeData () (see page 1547)	This is ProductTypeAttributeData, a member of class ProductTypeAttributeData.
	ProductTypeAttributeData.ProductTypeAttributeData (string, int, EkEnumeration.ProductTypeAttributeDataType, object, EkEnumeration.ProductTypeAttributeStatus) (see page 1547)	This is ProductTypeAttributeData, a member of class ProductTypeAttributeData.

ProductTypeAttributeData.ProductTypeAttributeData Constructor ()

C#

```
public ProductTypeAttributeData();
```

Visual Basic

```
Public Sub New()
```

Description

This is ProductTypeAttributeData, a member of class ProductTypeAttributeData.

ProductTypeAttributeData.ProductTypeAttributeData Constructor (string, int, EkEnumeration.ProductTypeAttributeDataType, object, EkEnumeration.ProductTypeAttributeStatus)

C#

```
public ProductTypeAttributeData(string name, int displayOrder,
EkEnumeration.ProductTypeAttributeDataType dataType, object defaultValue,
EkEnumeration.ProductTypeAttributeStatus activeStatus);
```

Visual Basic

```
Public Sub New(name As string, displayOrder As Integer, dataType As
EkEnumeration.ProductTypeAttributeDataType, defaultValue As object, activeStatus As
EkEnumeration.ProductTypeAttributeStatus)
```

Description

This is ProductTypeAttributeData, a member of class ProductTypeAttributeData.

ProductTypeAttributeData Properties

The properties of the ProductTypeAttributeData class are listed here.

Public Properties

	Name	Description
	ActiveStatus (see page 1548)	This is ActiveStatus, a member of class ProductTypeAttributeData.
	DataType (see page 1548)	This is DataType, a member of class ProductTypeAttributeData.
	DefaultValue (see page 1548)	This is DefaultValue, a member of class ProductTypeAttributeData.
	DisplayOrder (see page 1548)	This is DisplayOrder, a member of class ProductTypeAttributeData.
	Id (see page 1549)	This is Id, a member of class ProductTypeAttributeData.
	Name (see page 1549)	This is Name, a member of class ProductTypeAttributeData.

ProductTypeAttributeData.ActiveStatus Property

C#

```
public EkEnumeration.ProductTypeAttributeStatus ActiveStatus;
```

Visual Basic

```
Public Property ActiveStatus() As EkEnumeration.ProductTypeAttributeStatus
```

Description

This is ActiveStatus, a member of class ProductTypeAttributeData.

ProductTypeAttributeData.DataType Property

C#

```
public EkEnumeration.ProductTypeAttributeDataType DataType;
```

Visual Basic

```
Public Property DataType() As EkEnumeration.ProductTypeAttributeDataType
```

Description

This is DataType, a member of class ProductTypeAttributeData.

ProductTypeAttributeData.DefaultValue Property

C#

```
public object DefaultValue;
```

Visual Basic

```
Public Property DefaultValue() As object
```

Description

This is DefaultValue, a member of class ProductTypeAttributeData.

ProductTypeAttributeData.DisplayOrder Property

C#

```
public int DisplayOrder;
```

Visual Basic

```
Public Property DisplayOrder() As Integer
```

Description

This is DisplayOrder, a member of class ProductTypeAttributeData.

ProductTypeAttributeData.Id Property

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

Description

This is Id, a member of class ProductTypeAttributeData.

ProductTypeAttributeData.Name Property

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

Description

This is Name, a member of class ProductTypeAttributeData.

ProductTypeData Class

Data class representing ecommerce product types.

Class Hierarchy

C#

```
[Serializable]
public class ProductTypeData : XmlConfigData;
```

Visual Basic

```
<Serializable>
Public Class ProductTypeData
Inherits XmlConfigData
```

File

ProductTypeData.cs

ProductTypeData Properties

The properties of the ProductTypeData class are listed here.

Public Properties

	Name	Description
	Attributes (see page 1550)	This is Attributes, a member of class ProductTypeData.
	DefaultThumbnails (see page 1550)	This is DefaultThumbnails, a member of class ProductTypeData.
	EntryClass (see page 1550)	This is EntryClass, a member of class ProductTypeData.

ProductTypeData.Attributes Property

C#

```
public List<ProductTypeAttributeData> Attributes;
```

Visual Basic

```
Public Property Attributes() As List<ProductTypeAttributeData>
```

Description

This is Attributes, a member of class ProductTypeData.

ProductTypeData.DefaultThumbnails Property

C#

```
public List<ThumbnailDefaultData> DefaultThumbnails;
```

Visual Basic

```
Public Property DefaultThumbnails() As List<ThumbnailDefaultData>
```

Description

This is DefaultThumbnails, a member of class ProductTypeData.

ProductTypeData.EntryClass Property

C#

```
public EkEnumeration.CatalogEntryType EntryClass;
```

Visual Basic

```
Public Property EntryClass() As EkEnumeration.CatalogEntryType
```

Description

This is EntryClass, a member of class ProductTypeData.

RecommendationItemData Class

Data class representing entry reccomendations in ecommerce system.

Class Hierarchy

C#

```
public class RecommendationItemData : EntryGroupItemData;
```

Visual Basic

```
Public Class RecommendationItemData
Inherits EntryGroupItemData
```

File

RecommendationItemData.cs

RecommendationItemData Constructor

This is the overview for the RecommendationItemData constructor overload.

Overload List

	Name	Description
	RecommendationItemData.RecommendationItemData () (see page 1551)	This is RecommendationItemData, a member of class RecommendationItemData.
	RecommendationItemData.RecommendationItemData (long, long, long) (see page 1551)	This is RecommendationItemData, a member of class RecommendationItemData.
	RecommendationItemData.RecommendationItemData (long, long, long, EkEnumeration.RecommendationType) (see page 1551)	

RecommendationItemData.RecommendationItemData Constructor ()

C#

```
public RecommendationItemData();
```

Visual Basic

```
Public Sub New()
```

Description

This is RecommendationItemData, a member of class RecommendationItemData.

RecommendationItemData.RecommendationItemData Constructor (long, long, long)

C#

```
private RecommendationItemData(long id, long groupId, long entryId);
```

Visual Basic

```
Private Sub New(id As long, groupId As long, entryId As long)
```

Description

This is RecommendationItemData, a member of class RecommendationItemData.

RecommendationItemData.RecommendationItemData Constructor (long, long, long, EkEnumeration.RecommendationType)

C#

```
public RecommendationItemData(long id, long entryId, long entryLanguage,
EkEnumeration.RecommendationType recommendationType);
```

Visual Basic

```
Public Sub New(id As long, entryId As long, entryLanguage As long, recommendationType As
EkEnumeration.RecommendationType)
```

Parameters

Parameters	Description
long id	id of reccomendation.
long entryId	entry id associated with recommendation.
long entryLanguage	language of entryId.
EkEnumeration.RecommendationType recommendationType	type of recommendation.

RecommendationItemData Properties

The properties of the RecommendationItemData class are listed here.

Public Properties

	Name	Description
	EntryLanguage (see page 1552)	Gets or sets the language of the entry.
	EntryType (see page 1552)	Gets or sets the entry type associated with this recommendation.
	Html (see page 1552)	Gets or sets the Html of the recommendation.
	Image (see page 1553)	Gets or sets the image associated with the recommendation.
	ImageThumbnail (see page 1553)	Gets or sets the image thumbnail associated with the recommendation.
	Pricing (see page 1553)	The entry pricing, including currencies and quantity.
	Quicklink (see page 1553)	Gets or sets the quiclink of the recommendation.
	Summary (see page 1553)	Gets or sets the summary of the recommendation.
	Title (see page 1553)	Gets or sets the title of the recommendation.
	Type (see page 1554)	Gets or sets the type of reccomendation.

RecommendationItemData.EntryLanguage Property

Gets or sets the language of the entry.

C#

```
public long EntryLanguage;
```

Visual Basic

```
Public Property EntryLanguage() As long
```

RecommendationItemData.EntryType Property

Gets or sets the entry type associated with this recommendation.

C#

```
public EkEnumeration.CatalogEntryType EntryType;
```

Visual Basic

```
Public Property EntryType() As EkEnumeration.CatalogEntryType
```

RecommendationItemData.Html Property

Gets or sets the Html of the recommendation.

C#

```
public string Html;
```

Visual Basic

```
Public Property Html() As string
```

RecommendationItemData.Image Property

Gets or sets the image associated with the recommendation.

C#

```
public string Image;
```

Visual Basic

```
Public Property Image() As string
```

RecommendationItemData.ImageThumbnail Property

Gets or sets the image thumbnail associated with the recommendation.

C#

```
public string ImageThumbnail;
```

Visual Basic

```
Public Property ImageThumbnail() As string
```

RecommendationItemData.Pricing Property

The entry pricing, including currencies and quantity.

C#

```
public PricingData Pricing;
```

Visual Basic

```
Public Property Pricing() As PricingData
```

Returns

PricingData (see page 1540) of the catalog entry.

RecommendationItemData.Quicklink Property

Gets or sets the quiclink of the recommendation.

C#

```
public string Quicklink;
```

Visual Basic

```
Public Property Quicklink() As string
```

RecommendationItemData.Summary Property

Gets or sets the summary of the recommendation.

C#

```
public string Summary;
```

Visual Basic

```
Public Property Summary() As string
```

RecommendationItemData.Title Property

Gets or sets the title of the recommendation.

C#

```
public string Title;
```

Visual Basic

```
Public Property Title() As string
```

RecommendationItemData.Type Property

Gets or sets the type of reccomendation.

C#

```
public EkEnumeration.RecommendationType Type;
```

Visual Basic

```
Public Property Type() As EkEnumeration.RecommendationType
```

RegionData Class

Gets or sets information about specified geographic regions within countries that your eCommerce (see page 601) site services. For example, in the United States of America, each state is a region. So, if you were adding New Hampshire as a region, the code might look like this.

```
RegionData region = new RegionData();
region.Name = "New Hampshire";
region.CountryId = 840;   // 840 is the country code for the United States.
region.Code = "NH";
region.Enabled = true;
RegionAPI.Add(region);
```

To update an existing region's information, use the `Id` (see page 1557) property to specify the region. For example, to update New Hampshire's information, add `region.Id` (see page 1557) `= 31;` to the code above.

Class Hierarchy

Ektron.Cms.Commerce.RegionData

C#

```
public class RegionData;
```

Visual Basic

```
Public Class RegionData
```

File

RegionData.cs

RegionData Constructor

This is the overview for the RegionData constructor overload.

Overload List

	Name	Description
	RegionData.RegionData () (see page 1555)	This is RegionData, a member of class RegionData.

	RegionData.RegionData (string, long, string, bool) (see page 1555)	

RegionData.RegionData Constructor ()

C#

```
public RegionData();
```

Visual Basic

```
Public Sub New()
```

Description

This is RegionData, a member of class RegionData.

RegionData.RegionData Constructor (string, long, string, bool)

C#

```
public RegionData(string name, long countryId, string code, bool enabled);
```

Visual Basic

```
Public Sub New(name As string, countryId As long, code As string, enabled As bool)
```

Parameters

Parameters	Description
string name	The name of the region.
long countryId	The ID of the country to which the region belongs.
string code	Gets or sets the code of the region. This is typically an abbreviation for the name of the region.
bool enabled	Indicates whether or not the region is enabled for use.

RegionData Methods

The methods of the RegionData class are listed here.

Public Methods

	Name	Description
	ToString (see page 1555)	Returns Name (see page 1557) of Region.
	Validate (see page 1555)	Validates the RegionData (see page 1554) object is valid for saving.

RegionData.ToString Method

Returns Name (see page 1557) of Region.

C#

```
public override string ToString();
```

Visual Basic

```
Public override Function ToString() As string
```

RegionData.Validate Method

Validates the RegionData (see page 1554) object is valid for saving.

C#

```
public ValidationResults Validate();
```

Visual Basic

```
Public Function Validate() As ValidationResults
```

Returns

Returns a ValidationResult object with any validation errors.

RegionData Properties

The properties of the RegionData class are listed here.

Public Properties

	Name	Description
	Code (see page 1556)	Gets or sets the code of the region. This is typically an abbreviation for the name of the region. For example, if you add New Hampshire as a region, you would enter "NH" for this property. Each region must have its own unique code. You cannot have more than one region with the same code. This property is required when creating an region data object.
	CountryId (see page 1556)	Gets or sets the ID of the country to which the region belongs. For example, if you are adding a region to the United States, you would enter "840" for this property. This property is required when creating an region data object.
	Enabled (see page 1556)	Gets or sets whether or not the region is enabled for use.
	Id (see page 1557)	Gets or sets the ID of the region.
	Name (see page 1557)	Gets or sets the name of the region. For example, if the region is a US state, you might enter "New Hampshire" for this property. This property is required when creating an region data object.

RegionData.Code Property

Gets or sets the code of the region. This is typically an abbreviation for the name of the region. For example, if you add New Hampshire as a region, you would enter "NH" for this property. Each region must have its own unique code. You cannot have more than one region with the same code. This property is required when creating an region data object.

C#

```
public string Code;
```

Visual Basic

```
Public Property Code() As string
```

RegionData.CountryId Property

Gets or sets the ID of the country to which the region belongs. For example, if you are adding a region to the United States, you would enter "840" for this property. This property is required when creating an region data object.

C#

```
public long CountryId;
```

Visual Basic

```
Public Property CountryId() As long
```

RegionData.Enabled Property

Gets or sets whether or not the region is enabled for use.

C#

```
public bool Enabled;
```

Visual Basic

```
Public Property Enabled() As bool
```

RegionData.Id Property

Gets or sets the ID of the region.

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

RegionData.Name Property

Gets or sets the name of the region. For example, if the region is a US state, you might enter "New Hampshire" for this property. This property is required when creating an region data object.

C#

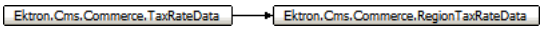
```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

RegionTaxRateData Class

Class Hierarchy



C#

```
public class RegionTaxRateData : TaxRateData;
```

Visual Basic

```
Public Class RegionTaxRateData
Inherits TaxRateData
```

File

RegionTaxRateData.cs

Description

This is class Ektron.Cms.Commerce.RegionTaxRateData.

RegionTaxRateData Constructor

This is the overview for the RegionTaxRateData constructor overload.

Overload List

	Name	Description
	RegionTaxRateData.RegionTaxRateData () (see page 1558)	This is RegionTaxRateData, a member of class RegionTaxRateData.
	RegionTaxRateData.RegionTaxRateData (long, long, decimal) (see page 1558)	This is RegionTaxRateData, a member of class RegionTaxRateData.

	RegionTaxRateData.RegionTaxRateData (long, long, decimal, long) (see page 1558)	This is RegionTaxRateData, a member of class RegionTaxRateData.

RegionTaxRateData.RegionTaxRateData Constructor ()

C#

```
public RegionTaxRateData();
```

Visual Basic

```
Public Sub New()
```

Description

This is RegionTaxRateData, a member of class RegionTaxRateData.

RegionTaxRateData.RegionTaxRateData Constructor (long, long, decimal)

C#

```
public RegionTaxRateData(long regionId, long taxClassId, decimal rate);
```

Visual Basic

```
Public Sub New(regionId As long, taxClassId As long, rate As decimal)
```

Description

This is RegionTaxRateData, a member of class RegionTaxRateData.

RegionTaxRateData.RegionTaxRateData Constructor (long, long, decimal, long)

C#

```
public RegionTaxRateData(long regionId, long taxClassId, decimal rate, long id);
```

Visual Basic

```
Public Sub New(regionId As long, taxClassId As long, rate As decimal, id As long)
```

Description

This is RegionTaxRateData, a member of class RegionTaxRateData.

RegionTaxRateData Properties

The properties of the RegionTaxRateData class are listed here.

Public Properties

	Name	Description
	RegionData (see page 1558)	Gets or sets the RegionData object associated with this tax rate.

RegionTaxRateData.RegionData Property

Gets or sets the RegionData object associated with this tax rate.

C#

```
public RegionData RegionData;
```

Visual Basic

```
Public Property RegionData() As RegionData
```

ShippingBaseData Class

Base data class for representing shipping method information.

Class Hierarchy

Ektron.Cms.Commerce.ShippingBaseData

C#

```
public class ShippingBaseData;
```

Visual Basic

```
Public Class ShippingBaseData
```

File

ShippingBaseData.cs

ShippingBaseData.ShippingBaseData Constructor

C#

```
public ShippingBaseData();
```

Visual Basic

```
Public Sub New()
```

Description

This is ShippingBaseData, a member of class ShippingBaseData.

ShippingBaseData Properties

The properties of the ShippingBaseData class are listed here.

Public Properties

	Name	Description
	Id (see page 1559)	gets or sets the id of the the shipping data.
	Name (see page 1560)	Gets or sets the name of the the shipping data
	ProviderService (see page 1560)	Gets or sets the name of the provider service for the shipping data.

ShippingBaseData.Id Property

gets or sets the id of the the shipping data.

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

ShippingBaseData.Name Property

Gets or sets the name of the the shipping data

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

ShippingBaseData.ProviderService Property

Gets or sets the name of the provider service for the shipping data.

C#

```
public string ProviderService;
```

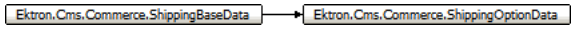
Visual Basic

```
Public Property ProviderService() As string
```

ShippingOptionData Class

Data class representing a quote for a particular shipping method.

Class Hierarchy



C#

```
public class ShippingOptionData : ShippingBaseData;
```

Visual Basic

```
Public Class ShippingOptionData
Inherits ShippingBaseData
```

File

ShippingFeeData.cs

ShippingOptionData.ShippingOptionData Constructor

C#

```
public ShippingOptionData();
```

Visual Basic

```
Public Sub New()
```

Description

This is ShippingOptionData, a member of class ShippingOptionData.

ShippingOptionData Properties

The properties of the ShippingOptionData class are listed here.

Public Properties

	Name	Description
	CurrencyId (see page 1561)	Gets or sets the currency associated with this shipping Fee.
	ShippingFee (see page 1561)	Gets or sets the fee for this particular shipping option.

ShippingOptionData.CurrencyId Property

Gets or sets the currency associated with this shipping Fee.

C#

```
public int CurrencyId;
```

Visual Basic

```
Public Property CurrencyId() As Integer
```

ShippingOptionData.ShippingFee Property

Gets or sets the fee for this particular shipping option.

C#

```
public decimal ShippingFee;
```

Visual Basic

```
Public Property ShippingFee() As decimal
```

TaxClassData Class

Data object representing tax class information.

Class Hierarchy

Ektron.Cms.Commerce.TaxClassData

C#

```
public class TaxClassData;
```

Visual Basic

```
Public Class TaxClassData
```

File

TaxClassData.cs

TaxClassData Constructor

This is the overview for the TaxClassData constructor overload.

Overload List

	Name	Description
	TaxClassData.TaxClassData () (see page 1562)	This is TaxClassData, a member of class TaxClassData.
	TaxClassData.TaxClassData (string) (see page 1562)	

	TaxClassData.TaxClassData (string, long) (see page 1562)	

TaxClassData.TaxClassData Constructor ()

C#

```
public TaxClassData();
```

Visual Basic

```
Public Sub New()
```

Description

This is TaxClassData, a member of class TaxClassData.

TaxClassData.TaxClassData Constructor (string)

C#

```
public TaxClassData(string name);
```

Visual Basic

```
Public Sub New(name As string)
```

Parameters

Parameters	Description
string name	name of the tax class

TaxClassData.TaxClassData Constructor (string, long)

C#

```
public TaxClassData(string name, long id);
```

Visual Basic

```
Public Sub New(name As string, id As long)
```

Parameters

Parameters	Description
string name	name of the tax class
long id	name of the tax class

TaxClassData Properties

The properties of the TaxClassData class are listed here.

Public Properties

	Name	Description
	Id (see page 1562)	Gets or sets the name of the tax class
	Name (see page 1563)	Gets or sets the name of the tax class.

TaxClassData.Id Property

Gets or sets the name of the tax class

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

TaxClassData.Name Property

Gets or sets the name of the tax class.

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

TaxRateData Class

Class Hierarchy

Ektron.Cms.Commerce.TaxRateData

C#

```
public class TaxRateData;
```

Visual Basic

```
Public Class TaxRateData
```

File

TaxRateData.cs

Description

This is class Ektron.Cms.Commerce.TaxRateData.

TaxRateData Constructor

This is the overview for the TaxRateData constructor overload.

Overload List

	Name	Description
	TaxRateData.TaxRateData () (see page 1564)	This is TaxRateData, a member of class TaxRateData.
	TaxRateData.TaxRateData (long, long, long, decimal) (see page 1564)	This is TaxRateData, a member of class TaxRateData.
	TaxRateData.TaxRateData (long, long, long, decimal, long) (see page 1564)	This is TaxRateData, a member of class TaxRateData.

TaxRateData.TaxRateData Constructor ()

C#

```
public TaxRateData();
```

Visual Basic

```
Public Sub New()
```

Description

This is TaxRateData, a member of class TaxRateData.

TaxRateData.TaxRateData Constructor (long, long, long, decimal)

C#

```
public TaxRateData(long typeId, long typeItemId, long taxClassId, decimal rate);
```

Visual Basic

```
Public Sub New(typeId As long, typeItemId As long, taxClassId As long, rate As decimal)
```

Description

This is TaxRateData, a member of class TaxRateData.

TaxRateData.TaxRateData Constructor (long, long, long, decimal, long)

C#

```
public TaxRateData(long typeId, long typeItemId, long taxClassId, decimal rate, long id);
```

Visual Basic

```
Public Sub New(typeId As long, typeItemId As long, taxClassId As long, rate As decimal, id As
long)
```

Description

This is TaxRateData, a member of class TaxRateData.

TaxRateData Properties

The properties of the TaxRateData class are listed here.

Public Properties

	Name	Description
	Id (see page 1564)	Gets or sets the Id of the TaxRate.
	Rate (see page 1565)	Gets or sets the actual rate for this TaxRate object. Should be entered in decimal format - i.e. .075 for 7.5%
	TaxClassId (see page 1565)	Gets or sets the Id (see page 1564) of the tax class that this rate applies to.
	TypeId (see page 1565)	Gets or sets the type of the Tax Rate (see page 1565).
	TypeItemId (see page 1565)	Gets or sets the Id (see page 1564) of the tax type object this tax rate applies to. For instance, for a region sales tax, TypeItemId would be the Id (see page 1564) of the corresponding region.

TaxRateData.Id Property

Gets or sets the Id of the TaxRate.

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

TaxRateData.Rate Property

Gets or sets the actual rate for this TaxRate object. Should be entered in decimal format - i.e. .075 for 7.5%

C#

```
public decimal Rate;
```

Visual Basic

```
Public Property Rate() As decimal
```

TaxRateData.TaxClassId Property

Gets or sets the Id (see page 1564) of the tax class that this rate applies to.

C#

```
public long TaxClassId;
```

Visual Basic

```
Public Property TaxClassId() As long
```

TaxRateData.TypeId Property

Gets or sets the type of the Tax Rate (see page 1565).

C#

```
public long TypeId;
```

Visual Basic

```
Public Property TypeId() As long
```

TaxRateData.TypeItemId Property

Gets or sets the Id (see page 1564) of the tax type object this tax rate applies to. For instance, for a region sales tax, TypeItemId would be the Id (see page 1564) of the corresponding region.

C#

```
public long TypeItemId;
```

Visual Basic

```
Public Property TypeItemId() As long
```

TaxTypeData Class

Data class representing Tax Types.

Class Hierarchy

Ektron.Cms.Commerce.TaxTypeData

C#

```
public class TaxTypeData;
```

Visual Basic

```
Public Class TaxTypeData
```

File

TaxTypeData.cs

TaxTypeData Constructor

This is the overview for the TaxTypeData constructor overload.

Overload List

	Name	Description
	TaxTypeData.TaxTypeData () (see page 1566)	This is TaxTypeData, a member of class TaxTypeData.
	TaxTypeData.TaxTypeData (string, int) (see page 1566)	
	TaxTypeData.TaxTypeData (string, int, long) (see page 1566)	

TaxTypeData.TaxTypeData Constructor ()

C#

```
public TaxTypeData();
```

Visual Basic

```
Public Sub New()
```

Description

This is TaxTypeData, a member of class TaxTypeData.

TaxTypeData.TaxTypeData Constructor (string, int)

C#

```
public TaxTypeData(string name, int precedence);
```

Visual Basic

```
Public Sub New(name As string, precedence As Integer)
```

Parameters

Parameters	Description
string name	name of the tax class
int precedence	Gets or sets the precedence of the tax class. This is the priority order in which it will be applied. For example, if type "Postal Code Sales Tax is priority 1" and it exists for an address it will be used over type "State Sales Tax" which is priority 2.

TaxTypeData.TaxTypeData Constructor (string, int, long)

C#

```
public TaxTypeData(string name, int precedence, long id);
```

Visual Basic

```
Public Sub New(name As string, precedence As Integer, id As long)
```

Parameters

Parameters	Description
string name	name of the tax class
int precedence	Gets or sets the precedence of the tax class. This is the priority order in which it will be applied. For example, if type "Postal Code Sales Tax is priority 1" and it exists for an address it will be used over type "State Sales Tax" which is priority 2.
long id	name of the tax class

TaxTypeData Properties

The properties of the TaxTypeData class are listed here.

Public Properties

	Name	Description
	Id (see page 1567)	Gets or sets the name of the tax class
	Name (see page 1567)	Gets or sets the name of the tax class.
	Precedence (see page 1567)	Gets or sets the precedence of the tax class. This is the priority order in which it will be applied. For example, if type "Postal Code Sales Tax is priority 1" and it exists for an address it will be used over type "State Sales Tax" which is priority 2.

TaxTypeData.Id Property

Gets or sets the name of the tax class

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

TaxTypeData.Name Property

Gets or sets the name of the tax class.

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

TaxTypeData.Precedence Property

Gets or sets the precedence of the tax class. This is the priority order in which it will be applied. For example, if type "Postal Code Sales Tax is priority 1" and it exists for an address it will be used over type "State Sales Tax" which is priority 2.

C#

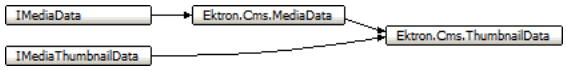
```
public int Precedence;
```

Visual Basic

```
Public Property Precedence() As Integer
```

ThumbnailData Class

Class Hierarchy



C#

```
public class ThumbnailData : MediaData, IMediaThumbnailData;
```

Visual Basic

```
Public Class ThumbnailData
Inherits MediaData
Inherits IMediaThumbnailData
```

File

ThumbnailData.cs

Description

This is class Ektron.Cms.ThumbnailData.

ThumbnailData Properties

The properties of the ThumbnailData class are listed here.

Public Properties

	Name	Description
	Height (see page 1568)	This is Height, a member of class ThumbnailData.
	Width (see page 1568)	This is Width, a member of class ThumbnailData.

ThumbnailData.Height Property

C#

```
public int Height;
```

Visual Basic

```
Public Property Height() As Integer
```

Description

This is Height, a member of class ThumbnailData.

ThumbnailData.Width Property

C#

```
public int Width;
```

Visual Basic

```
Public Property Width() As Integer
```

Description

This is Width, a member of class ThumbnailData.

ThumbnailDefaultData Class

Class Hierarchy

C#

```
public class ThumbnailDefaultData : IMediaThumbnailDefault;
```

Visual Basic

```
Public Class ThumbnailDefaultData
Inherits IMediaThumbnailDefault
```

File

ThumbnailDefaultData.cs

Description

This is class Ektron.Cms.ThumbnailDefaultData.

ThumbnailDefaultData Properties

The properties of the ThumbnailDefaultData class are listed here.

Public Properties

	Name	Description
	FileNameFormat (see page 1569)	This is FileNameFormat, a member of class ThumbnailDefaultData.
	Height (see page 1569)	This is Height, a member of class ThumbnailDefaultData.
	Id (see page 1570)	This is Id, a member of class ThumbnailDefaultData.
	MimeType (see page 1570)	This is MimeType, a member of class ThumbnailDefaultData.
	Title (see page 1570)	This is Title, a member of class ThumbnailDefaultData.
	Width (see page 1570)	This is Width, a member of class ThumbnailDefaultData.

ThumbnailDefaultData.FileNameFormat Property

C#

```
public string FileNameFormat;
```

Visual Basic

```
Public Property FileNameFormat() As string
```

Description

This is FileNameFormat, a member of class ThumbnailDefaultData.

ThumbnailDefaultData.Height Property

C#

```
public int Height;
```

Visual Basic

```
Public Property Height() As Integer
```

Description

This is Height, a member of class ThumbnailDefaultData.

ThumbnailDefaultData.Id Property

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

Description

This is Id, a member of class ThumbnailDefaultData.

ThumbnailDefaultData.MimeType Property

C#

```
public string MimeType;
```

Visual Basic

```
Public Property MimeType() As string
```

Description

This is MimeType, a member of class ThumbnailDefaultData.

ThumbnailDefaultData.Title Property

C#

```
public string Title;
```

Visual Basic

```
Public Property Title() As string
```

Description

This is Title, a member of class ThumbnailDefaultData.

ThumbnailDefaultData.Width Property

C#

```
public int Width;
```

Visual Basic

```
Public Property Width() As Integer
```

Description

This is Width, a member of class ThumbnailDefaultData.

TierPriceData Class

Class Hierarchy

C#

```
public class TierPriceData : CmsData<TierPriceData>, IComparable<TierPriceData>;
```

Visual Basic

```
Public Class TierPriceData
Inherits CmsData
Inherits IComparable
```

File

TierPriceData.cs

Description

This is class Ektron.Cms.Commerce.TierPriceData.

TierPriceData Constructor

This is the overview for the TierPriceData constructor overload.

Overload List

	Name	Description
	TierPriceData.TierPriceData () (see page 1571)	This is TierPriceData, a member of class TierPriceData.
	TierPriceData.TierPriceData (int, decimal) (see page 1571)	This is TierPriceData, a member of class TierPriceData.
	TierPriceData.TierPriceData (int, decimal, long) (see page 1572)	This is TierPriceData, a member of class TierPriceData.

TierPriceData.TierPriceData Constructor ()

C#

```
public TierPriceData();
```

Visual Basic

```
Public Sub New()
```

Description

This is TierPriceData, a member of class TierPriceData.

TierPriceData.TierPriceData Constructor (int, decimal)

C#

```
public TierPriceData(int quantity, decimal salePrice);
```

Visual Basic

```
Public Sub New(quantity As Integer, salePrice As decimal)
```

Description

This is TierPriceData, a member of class TierPriceData.

TierPriceData.TierPriceData Constructor (int, decimal, long)

C#

```
public TierPriceData(int quantity, decimal salePrice, long id);
```

Visual Basic

```
Public Sub New(quantity As Integer, salePrice As decimal, id As long)
```

Description

This is TierPriceData, a member of class TierPriceData.

TierPriceData Methods

The methods of the TierPriceData class are listed here.

Public Methods

	Name	Description
	Clone (see page 1572)	This is Clone, a member of class TierPriceData.
	CompareTo (see page 1572)	This is CompareTo, a member of class TierPriceData.
	Validate (see page 1572)	This is Validate, a member of class TierPriceData.

TierPriceData.Clone Method

C#

```
public TierPriceData Clone();
```

Visual Basic

```
Public Function Clone() As TierPriceData
```

Description

This is Clone, a member of class TierPriceData.

TierPriceData.CompareTo Method

C#

```
public int CompareTo(TierPriceData other);
```

Visual Basic

```
Public Function CompareTo(other As TierPriceData) As Integer
```

Description

This is CompareTo, a member of class TierPriceData.

TierPriceData.Validate Method

C#

```
public override ValidationResults Validate();
```

Visual Basic

```
Public override Function Validate() As ValidationResults
```

Description

This is Validate, a member of class TierPriceData.

TierPriceData Properties

The properties of the TierPriceData class are listed here.

Public Properties

	Name	Description
	DataState (see page 1573)	Gets or sets the TierPrices persistence State. Used internally.
	Id (see page 1573)	This is Id, a member of class TierPriceData.
	Quantity (see page 1573)	This is Quantity, a member of class TierPriceData.
	RecurrenceId (see page 1573)	Gets or sets the id of the recurrence pattern for this price if it supports recurring billing.
	SalePrice (see page 1574)	This is SalePrice, a member of class TierPriceData.

TierPriceData.DataState Property

Gets or sets the TierPrices persistence State. Used internally.

C#

```
public EkEnumeration.DataState DataState;
```

Visual Basic

```
Public Property DataState() As EkEnumeration.DataState
```

TierPriceData.Id Property

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

Description

This is Id, a member of class TierPriceData.

TierPriceData.Quantity Property

C#

```
[RangeValidator(1, RangeBoundaryType.Inclusive, 0, RangeBoundaryType.Ignore,
MessageTemplateResourceName="TierPricing_QuantityNegative",
MessageTemplateResourceType=typeof(Errors))]
public int Quantity;
```

Visual Basic

```
<RangeValidator(1, RangeBoundaryType.Inclusive, 0, RangeBoundaryType.Ignore,
MessageTemplateResourceName="TierPricing_QuantityNegative",
MessageTemplateResourceType=typeof(Errors))>
Public Property Quantity() As Integer
```

Description

This is Quantity, a member of class TierPriceData.

TierPriceData.RecurrenceId Property

Gets or sets the id of the recurrence pattern for this price if it supports recurring billing.

C#

```
public long RecurrenceId;
```

Visual Basic

```
Public Property RecurrenceId() As long
```

TierPriceData.SalePrice Property

C#

```
public decimal SalePrice;
```

Visual Basic

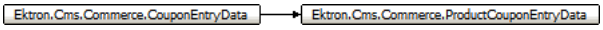
```
Public Property SalePrice() As decimal
```

Description

This is SalePrice, a member of class TierPriceData.

ProductCouponEntryData Class

Class Hierarchy



C#

```
public class ProductCouponEntryData : CouponEntryData;
```

Visual Basic

```
Public Class ProductCouponEntryData
Inherits CouponEntryData
```

File

CouponEntryData.cs

Description

This is class Ektron.Cms.Commerce.ProductCouponEntryData.

ProductCouponEntryData Constructor

This is the overview for the ProductCouponEntryData constructor overload.

Overload List

	Name	Description
	ProductCouponEntryData.ProductCouponEntryData () (see page 1575)	This is ProductCouponEntryData, a member of class ProductCouponEntryData.
	ProductCouponEntryData.ProductCouponEntryData (long, EkEnumeration.CMSObjectTypes, long, string, bool, EkEnumeration.CatalogEntryType) (see page 1575)	This is ProductCouponEntryData, a member of class ProductCouponEntryData.

ProductCouponEntryData.ProductCouponEntryData Constructor ()

C#

```
public ProductCouponEntryData();
```

Visual Basic

```
Public Sub New()
```

Description

This is ProductCouponEntryData, a member of class ProductCouponEntryData.

ProductCouponEntryData.ProductCouponEntryData Constructor (long, EkEnumeration.CMSObjectTypes, long, string, bool, EkEnumeration.CatalogEntryType)

C#

```
public ProductCouponEntryData(long objectId, EkEnumeration.CMSObjectTypes objectType, long
couponId, string title, bool isIncluded, EkEnumeration.CatalogEntryType entryType);
```

Visual Basic

```
Public Sub New(objectId As long, objectType As EkEnumeration.CMSObjectTypes, couponId As long,
title As string, isIncluded As bool, entryType As EkEnumeration.CatalogEntryType)
```

Description

This is ProductCouponEntryData, a member of class ProductCouponEntryData.

ProductCouponEntryData Properties

The properties of the ProductCouponEntryData class are listed here.

Public Properties

	Name	Description
	EntryType (see page 1575)	Gets or sets the type of catalog entry.

ProductCouponEntryData.EntryType Property

Gets or sets the type of catalog entry.

C#

```
public Ektron.Cms.Common.EkEnumeration.CatalogEntryType EntryType;
```

Visual Basic

```
Public Property EntryType() As Ektron.Cms.Common.EkEnumeration.CatalogEntryType
```

WarehouseData Class

Class Hierarchy

Ektron.Cms.Commerce.WarehouseData

C#

```
public class WarehouseData;
```

Visual Basic

```
Public Class WarehouseData
```

File

WarehouseData.cs

Description

This is class Ektron.Cms.Commerce.WarehouseData.

WarehouseData Constructor

This is the overview for the WarehouseData constructor overload.

Overload List

	Name	Description
	WarehouseData.WarehouseData () (see page 1576)	This is WarehouseData, a member of class WarehouseData.
	WarehouseData.WarehouseData (string, AddressData) (see page 1576)	This is WarehouseData, a member of class WarehouseData.

WarehouseData.WarehouseData Constructor ()

C#

```
public WarehouseData();
```

Visual Basic

```
Public Sub New()
```

Description

This is WarehouseData, a member of class WarehouseData.

WarehouseData.WarehouseData Constructor (string, AddressData)

C#

```
public WarehouseData(string warehouseName, AddressData address);
```

Visual Basic

```
Public Sub New(warehouseName As string, address As AddressData)
```

Description

This is WarehouseData, a member of class WarehouseData.

WarehouseData Properties

The properties of the WarehouseData class are listed here.

Public Properties

	Name	Description
	Address (see page 1577)	Gets or sets the address of this Warehouse.
	Id (see page 1577)	Gets or sets Id of Address (see page 1577).
	IsDefaultWarehouse (see page 1577)	Gets or sets the IsDefaultWarehouse status property for the warehouse. If true, this warehouse is the default warehouse for the system.
	Name (see page 1577)	Gets or sets the name of this Warehouse.

WarehouseData.Address Property

Gets or sets the address of this Warehouse.

C#

```
public AddressData Address;
```

Visual Basic

```
Public Property Address() As AddressData
```

WarehouseData.Id Property

Gets or sets Id of Address (see page 1577).

C#

```
public long Id;
```

Visual Basic

```
Public Property Id() As long
```

WarehouseData.IsDefaultWarehouse Property

Gets or sets the IsDefaultWarehouse status property for the warehouse. If true, this warehouse is the default warehouse for the system.

C#

```
public bool IsDefaultWarehouse;
```

Visual Basic

```
Public Property IsDefaultWarehouse() As bool
```

WarehouseData.Name Property

Gets or sets the name of this Warehouse.

C#

```
public string Name;
```

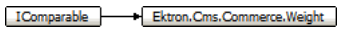
Visual Basic

```
Public Property Name() As string
```

Weight Class

Class used to define mesurement information.

Class Hierarchy



C#

```
public class Weight : IComparable<Weight>;
```

Visual Basic

```
Public Class Weight
Inherits IComparable
```

File

Measure.cs

Weight Constructor

This is the overview for the Weight constructor overload.

Overload List

	Name	Description
	Weight.Weight () (see page 1578)	This is Weight, a member of class Weight.
	Weight.Weight (WeightUnit, float) (see page 1578)	

Weight.Weight Constructor ()

C#

```
public Weight();
```

Visual Basic

```
Public Sub New()
```

Description

This is Weight, a member of class Weight.

Weight.Weight Constructor (WeightUnit, float)

C#

```
public Weight(WeightUnit units, float totalWeight);
```

Visual Basic

```
Public Sub New(units As WeightUnit, totalWeight As float)
```

Parameters

Parameters	Description
WeightUnit units	Units (see page 1580) in which the package is weighed.
float totalWeight	Total weight of the package in Units (see page 1580).

Weight Methods

The methods of the Weight class are listed here.

Public Methods

	Name	Description
	Clone (see page 1579)	This is Clone, a member of class Weight.
	CompareTo (see page 1579)	This is CompareTo, a member of class Weight.
	ConvertUnits (see page 1579)	Converts the current Weight (see page 1577) object to a new object in the supplied units.

Weight.Clone Method

C#

```
public Weight Clone();
```

Visual Basic

```
Public Function Clone() As Weight
```

Description

This is Clone, a member of class Weight.

Weight.CompareTo Method

C#

```
public int CompareTo(Weight other);
```

Visual Basic

```
Public Function CompareTo(other As Weight) As Integer
```

Description

This is CompareTo, a member of class Weight.

Weight.ConvertUnits Method

Converts the current Weight (see page 1577) object to a new object in the supplied units.

C#

```
public Weight ConvertUnits(WeightUnit toUnit);
```

Visual Basic

```
Public Function ConvertUnits(toUnit As WeightUnit) As Weight
```

Parameters

Parameters	Description
WeightUnit toUnit	WeightUnit (see page 1588) to convert to

Weight Properties

The properties of the Weight class are listed here.

Public Properties

	Name	Description
	Amount (see page 1579)	Gets or sets the actual weight value in Units (see page 1580).
	Units (see page 1580)	Gets or sets the Units in which the package is weighed.

Weight.Amount Property

Gets or sets the actual weight value in Units (see page 1580).

C#

```
public float Amount;
```

Visual Basic

```
Public Property Amount() As float
```

Weight.Units Property

Gets or sets the Units in which the package is weighed.

C#

```
public WeightUnit Units;
```

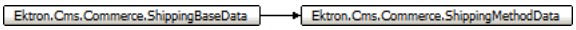
Visual Basic

```
Public Property Units() As WeightUnit
```

ShippingMethodData Class

Data class used to represent shipping methods defined in the cms.

Class Hierarchy



C#

```
public class ShippingMethodData : ShippingBaseData;
```

Visual Basic

```
Public Class ShippingMethodData
Inherits ShippingBaseData
```

File

ShippingData.cs

ShippingMethodData.ShippingMethodData Constructor

C#

```
public ShippingMethodData();
```

Visual Basic

```
Public Sub New()
```

Description

This is ShippingMethodData, a member of class ShippingMethodData.

ShippingMethodData Properties

The properties of the ShippingMethodData class are listed here.

Public Properties

	Name	Description
	DisplayOrder (see page 1580)	Gets or sets the display order for this shipping method.
	IsActive (see page 1581)	Gets or sets whethe ror not the Shipping Method is Active.

ShippingMethodData.DisplayOrder Property

Gets or sets the display order for this shipping method.

C#

```
public int DisplayOrder;
```

Visual Basic

```
Public Property DisplayOrder() As Integer
```

ShippingMethodData.IsActive Property

Gets or sets whethe ror not the Shipping Method is Active.

C#

```
public bool IsActive;
```

Visual Basic

```
Public Property IsActive() As bool
```

Structs, Records, Enums

The following table lists structs, records, enums in this documentation.

Enumerations

	Name	Description
	AddressProperty (see page 1582)	Enumeration of Address properties used for sorting and filtering AddressData (see page 1428) objects.
	CountryProperty (see page 1582)	Enumeration of CountryData (see page 1461) properties used for sorting and filtering CountryData (see page 1461) objects.
	CouponProperty (see page 1582)	Enumeration of Coupon properties used for sorting and filtering CouponData (see page 1466) objects.
	CurrencyProperty (see page 1583)	Enumeration of CurrencyData (see page 1472) properties used for sorting and filtering CurrencyData (see page 1472) objects.
	CustomerProperty (see page 1583)	Enumeration of Customer properties used for sorting and filtering CustomerData (see page 1478) objects.
	EntryAttributeProperty (see page 1583)	Enumeration of EntryAttributeProperty properties available for sorting and filtering.
	EntryGroupItemProperty (see page 1583)	Enumeration of Entry Group Item properties used for sorting and filtering EntryGroupItemData (see page 1501) objects.
	EntryGroupProperty (see page 1584)	Enumeration of EntryGroup properties used for sorting and filtering EntryGroupData (see page 1499) objects.
	EntryProperty (see page 1584)	Enumeration of EntryData (see page 1487) columns available for sorting and filtering.
	ExchangeRateProperty (see page 1584)	Enumeration of ExchangeRate columns available for sorting and filtering.
	LinearUnit (see page 1585)	Enumeration of units for measuring length.
	OrderProperty (see page 1585)	Enumeration of order properties used for sorting and filtering OrderData (see page 1520) objects.
	PackageProperty (see page 1585)	Enumeration representing properties of the packagedata class.
	ProductTypeAttributeProperty (see page 1586)	Enumeration of ProductTypeAttributeData (see page 1546) properties available for sorting and filtering.
	ProductTypeProperty (see page 1586)	Enumeration of ProductType properties used for sorting and filtering CustomerData (see page 1478) objects.
	RegionProperty (see page 1586)	Enumeration of Region properties used for sorting and filtering RegionData (see page 1554) objects.
	ShippingMethodProperty (see page 1586)	Enumeration representing properties of the ShippingMethodData (see page 1580) class.
	TaxClassProperty (see page 1587)	/ Enumeration of TaxClass columns available for sorting and filtering.

	TaxRateProperty (see page 1587)	Enumeration of possible values to order Lists of TaxRateData (see page 1563) by.
	TaxRateType (see page 1587)	This Represents the default Tax rate Types supported by The CMS. Additonal tax Types may be supported by customers.
	TaxTypeProperty (see page 1588)	/ Enumeration of TaxType columns available for sorting and filtering.
	WarehouseProperty (see page 1588)	Enumeration of Warehouse properties used for sorting and filtering WarehouseData (see page 1575) objects.
	WeightUnit (see page 1588)	Enumeration of units for measuring weight.

Ektron.Cms.Commerce.AddressProperty Enumeration

Enumeration of Address properties used for sorting and filtering AddressData (see page 1428) objects.

C#

```
public enum AddressProperty {
}
```

Visual Basic

```
Public Enum AddressProperty
End Enum
```

File

AddressData.cs

Ektron.Cms.Commerce.CountryProperty Enumeration

Enumeration of CountryData (see page 1461) properties used for sorting and filtering CountryData (see page 1461) objects.

C#

```
public enum CountryProperty {
}
```

Visual Basic

```
Public Enum CountryProperty
End Enum
```

File

CountryData.cs

Ektron.Cms.Commerce.CouponProperty Enumeration

Enumeration of Coupon properties used for sorting and filtering CouponData (see page 1466) objects.

C#

```
public enum CouponProperty {
}
```

Visual Basic

```
Public Enum CouponProperty
End Enum
```

File

CouponData.cs

Ektron.Cms.Commerce.CurrencyProperty Enumeration

Enumeration of CurrencyData (see page 1472) properties used for sorting and filtering CurrencyData (see page 1472) objects.

C#

```
public enum CurrencyProperty : int {
}
```

Visual Basic

```
Public Enum CurrencyProperty
Inherits int
End Enum
```

File

CurrencyData.cs

Ektron.Cms.Commerce.CustomerProperty Enumeration

Enumeration of Customer properties used for sorting and filtering CustomerData (see page 1478) objects.

C#

```
public enum CustomerProperty {
}
```

Visual Basic

```
Public Enum CustomerProperty
End Enum
```

File

CustomerData.cs

Ektron.Cms.Commerce.EntryAttributeProperty Enumeration

Enumeration of EntryAttributeProperty properties available for sorting and filtering.

C#

```
public enum EntryAttributeProperty : int {
}
```

Visual Basic

```
Public Enum EntryAttributeProperty
Inherits int
End Enum
```

File

ProductAttributeData.cs

Ektron.Cms.Commerce.EntryGroupItemProperty Enumeration

Enumeration of Entry Group Item properties used for sorting and filtering EntryGroupItemData (see page 1501) objects.

C#

```
public enum EntryGroupItemProperty : int {
}
```

Visual Basic

```
Public Enum EntryGroupItemProperty
Inherits int
End Enum
```

File

EntryGroupItemData.cs

Ektron.Cms.Commerce.EntryGroupProperty Enumeration

Enumeration of EntryGroup properties used for sorting and filtering EntryGroupData (see page 1499) objects.

C#

```
public enum EntryGroupProperty : int {
}
```

Visual Basic

```
Public Enum EntryGroupProperty
Inherits int
End Enum
```

File

EntryGroupData.cs

Ektron.Cms.Commerce.EntryProperty Enumeration

Enumeration of EntryData (see page 1487) columns available for sorting and filtering.

C#

```
public enum EntryProperty {
}
```

Visual Basic

```
Public Enum EntryProperty
End Enum
```

File

EntryData.cs

Ektron.Cms.Commerce.ExchangeRateProperty Enumeration

Enumeration of ExchangeRate columns available for sorting and filtering.

C#

```
public enum ExchangeRateProperty {
}
```

Visual Basic

```
Public Enum ExchangeRateProperty
```

```
End Enum
```

File

ExchangeRateData.cs

Ektron.Cms.Commerce.LinearUnit Enumeration

Enumeration of units for measuring length.

C#

```
public enum LinearUnit {
}
```

Visual Basic

```
Public Enum LinearUnit
End Enum
```

File

Measure.cs

Ektron.Cms.Commerce.OrderProperty Enumeration

Enumeration of order properties used for sorting and filtering OrderData (see page 1520) objects.

C#

```
public enum OrderProperty : int {
}
```

Visual Basic

```
Public Enum OrderProperty
Inherits int
End Enum
```

File

OrderData.cs

Ektron.Cms.Commerce.PackageProperty Enumeration

Enumeration representing properties of the packagedata class.

C#

```
public enum PackageProperty {
}
```

Visual Basic

```
Public Enum PackageProperty
End Enum
```

File

PackageData.cs

Ektron.Cms.Commerce.ProductTypeAttributeProperty Enumeration

Enumeration of ProductTypeAttributeData (see page 1546) properties available for sorting and filtering.

C#

```
public enum ProductTypeAttributeProperty : int {
}
```

Visual Basic

```
Public Enum ProductTypeAttributeProperty
Inherits int
End Enum
```

File

ProductAttributeData.cs

Ektron.Cms.Commerce.ProductTypeProperty Enumeration

Enumeration of ProductType properties used for sorting and filtering CustomerData (see page 1478) objects.

C#

```
public enum ProductTypeProperty : int {
}
```

Visual Basic

```
Public Enum ProductTypeProperty
Inherits int
End Enum
```

File

ProductTypeData.cs

Ektron.Cms.Commerce.RegionProperty Enumeration

Enumeration of Region properties used for sorting and filtering RegionData (see page 1554) objects.

C#

```
public enum RegionProperty : int {
}
```

Visual Basic

```
Public Enum RegionProperty
Inherits int
End Enum
```

File

RegionData.cs

Ektron.Cms.Commerce.ShippingMethodProperty Enumeration

Enumeration representing properties of the ShippingMethodData (see page 1580) class.

C#

```
public enum ShippingMethodProperty {
}
```

Visual Basic

```
Public Enum ShippingMethodProperty
End Enum
```

File

ShippingData.cs

Ektron.Cms.Commerce.TaxClassProperty Enumeration

/ Enumeration of TaxClass columns available for sorting and filtering.

C#

```
public enum TaxClassProperty {
}
```

Visual Basic

```
Public Enum TaxClassProperty
End Enum
```

File

TaxClassData.cs

Ektron.Cms.Commerce.TaxRateProperty Enumeration

Enumeration of possible values to order Lists of TaxRateData (see page 1563) by.

C#

```
public enum TaxRateProperty {
}
```

Visual Basic

```
Public Enum TaxRateProperty
End Enum
```

File

TaxRateData.cs

Ektron.Cms.Commerce.TaxRateType Enumeration

This Represents the default Tax rate Types supported by The CMS. Additonal tax Types may be supported by customers.

C#

```
public enum TaxRateType {
}
```

Visual Basic

```
Public Enum TaxRateType
End Enum
```

File

TaxRateData.cs

Remarks

TaxRateType Id's 1-100 should be reserved for internal use.

Example

customer could create specialized VAT TaxRate type.

Ektron.Cms.Commerce.TaxTypeProperty Enumeration

/ Enumeration of TaxType columns available for sorting and filtering.

C#

```
public enum TaxTypeProperty {
}
```

Visual Basic

```
Public Enum TaxTypeProperty
End Enum
```

File

TaxTypeData.cs

Ektron.Cms.Commerce.WarehouseProperty Enumeration

Enumeration of Warehouse properties used for sorting and filtering WarehouseData (see page 1575) objects.

C#

```
public enum WarehouseProperty : int {
}
```

Visual Basic

```
Public Enum WarehouseProperty
Inherits int
End Enum
```

File

WarehouseData.cs

Ektron.Cms.Commerce.WeightUnit Enumeration

Enumeration of units for measuring weight.

C#

```
public enum WeightUnit {
}
```

Visual Basic

```
Public Enum WeightUnit
End Enum
```

File

Measure.cs

Ektron.Cms.Common Namespace

This is namespace Ektron.Cms.Common.

Classes

	Name	Description
	CommerceSettings (see page 1589)	This is class Ektron.Cms.Common.CommerceSettings.
	UrlAliasAutoData (see page 1592)	A data class used to get or set information about an Automatic URL Alias.
	UrlAliasInfo (see page 1597)	This is class Ektron.Cms.Common.UrlAliasInfo.
	UrlAliasManualData (see page 1599)	A data class used to get or set information about a Manual URL Alias.
	UrlAliasRegExData (see page 1604)	A data class used to get or set information about a URL Alias that uses RegEx.

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	CommerceSettings (see page 1589)	This is class Ektron.Cms.Common.CommerceSettings.
	UrlAliasAutoData (see page 1592)	A data class used to get or set information about an Automatic URL Alias.
	UrlAliasInfo (see page 1597)	This is class Ektron.Cms.Common.UrlAliasInfo.
	UrlAliasManualData (see page 1599)	A data class used to get or set information about a Manual URL Alias.
	UrlAliasRegExData (see page 1604)	A data class used to get or set information about a URL Alias that uses RegEx.

CommerceSettings Class

Class Hierarchy

Ektron.Cms.Common.CommerceSettings

C#

```
[Serializable()]
public class CommerceSettings;
```

Visual Basic

```
<Serializable()>
Public Class CommerceSettings
```

File

EkRequestInformation.vb

Description

This is class Ektron.Cms.Common.CommerceSettings.

CommerceSettings Properties

The properties of the CommerceSettings class are listed here.

Public Properties

	Name	Description
	ComplianceMode (see page 1590)	Gets or sets compliance mode.
	CurrencyAlphaISO (see page 1590)	Gets or sets the Alpha ISO code the current currency.
	CurrencyCultureCode (see page 1590)	Gets or sets the culture code associated with the current currency.
	CurrencyId (see page 1591)	Gets or sets the current currency Id for the request.
R	CurrencySymbol (see page 1591)	Read only function that gets the ISO symbol for this currency.
	DefaultCurrencyId (see page 1591)	Gets or sets the default currency Id.
R	ISOCurrencySymbol (see page 1591)	Read only function that gets the symbol for this currency.
	OrderProcessingDisabled (see page 1592)	Gets or sets whether order processing is disabled (eSynch).
	PasswordHistory (see page 1592)	Gets or sets the number of previous passwords to store for a user.
	TestMode (see page 1592)	Gets or sets the default currency Id.

CommerceSettings.ComplianceMode Property

Gets or sets compliance mode.

C#

```
public Boolean ComplianceMode;
```

Visual Basic

```
Public Property ComplianceMode() As Boolean
```

Returns

Boolean

CommerceSettings.CurrencyAlphaISO Property

Gets or sets the Alpha ISO code the current currency.

C#

```
public String CurrencyAlphaISO;
```

Visual Basic

```
Public Property CurrencyAlphaISO() As String
```

CommerceSettings.CurrencyCultureCode Property

Gets or sets the culture code associated with the current currency.

C#

```
public String CurrencyCultureCode;
```

Visual Basic

```
Public Property CurrencyCultureCode() As String
```

CommerceSettings.CurrencyId Property

Gets or sets the current currency Id for the request.

C#

```
public int CurrencyId;
```

Visual Basic

```
Public Property CurrencyId() As Integer
```

CommerceSettings.CurrencySymbol Property

Read only function that gets the ISO symbol for this currency.

C#

```
public ReadOnly String CurrencySymbol;
```

Visual Basic

```
Public ReadOnly Property CurrencySymbol() As String
```

Returns

ISO symbol for the currency, for example USD

Example

USD

CommerceSettings.DefaultCurrencyId Property

Gets or sets the default currency Id.

C#

```
public int DefaultCurrencyId;
```

Visual Basic

```
Public Property DefaultCurrencyId() As Integer
```

CommerceSettings.ISOCurrencySymbol Property

Read only function that gets the symbol for this currency.

C#

```
public ReadOnly String ISOCurrencySymbol;
```

Visual Basic

```
Public ReadOnly Property ISOCurrencySymbol() As String
```

Returns

Currency symbol for the currency, for example $

Example

$

CommerceSettings.OrderProcessingDisabled Property

Gets or sets whether order processing is disabled (eSynch).

C#

```
public Boolean OrderProcessingDisabled;
```

Visual Basic

```
Public Property OrderProcessingDisabled() As Boolean
```

Returns

Boolean

CommerceSettings.PasswordHistory Property

Gets or sets the number of previous passwords to store for a user.

C#

```
public int PasswordHistory;
```

Visual Basic

```
Public Property PasswordHistory() As Integer
```

Returns

Boolean

CommerceSettings.TestMode Property

Gets or sets the default currency Id.

C#

```
public Boolean TestMode;
```

Visual Basic

```
Public Property TestMode() As Boolean
```

UrlAliasAutoData Class

A data class used to get or set information about an Automatic URL Alias.

Class Hierarchy

Ektron.Cms.Common.UrlAliasAutoData

C#

```
public class UrlAliasAutoData;
```

Visual Basic

```
Public Class UrlAliasAutoData
```

File

UrlAliasAutoData.vb

UrlAliasAutoData Properties

The properties of the UrlAliasAutoData class are listed here.

Public Properties

	Name	Description
	AliasName (see page 1593)	Gets or sets the alias name for the URL.
	AutoAliasType (see page 1593)	Gets or sets the automatic alias source type. Types are Taxonomy (see page 876) and Folder (see page 683).
	AutoId (see page 1594)	Gets the unique ID generated for this record.
	Example (see page 1594)	Gets or sets a sample of the selected configuration.
	ExcludedPath (see page 1594)	Returns the folder or taxonomy path excluded from the selected path.
	FileExtension (see page 1594)	Gets or sets the selected file extension.
R	FormattedSelectedPath (see page 1595)	Returns the folder or taxonomy path with modifications
	HostName (see page 1595)	Gets the name of the host where the content resides.
	IsEnabled (see page 1595)	Gets or sets whether the configuration is active or not.
	LanguageId (see page 1595)	Gets the language ID for the automatic alias.
	PageNameType (see page 1595)	Gets or sets the name for the configuration. Available types are: • ContentTitle • ContentId • ContentIdAndLanguage
	ReplacementCharacter (see page 1596)	Gets or sets the character that will replace special characters in the generated URL.
	SelectedPath (see page 1596)	Returns the folder or taxonomy path selected by the user
	SiteId (see page 1596)	Gets or sets the root folder ID of the site. If no site is specified, returns 0 (zero).
	SourceId (see page 1596)	Gets or sets either the taxonomy ID or the folder ID depending on the auto alias type selected.
	SourceName (see page 1597)	Gets the name of the taxonomy or folder selected depending on the auto alias type selected.
	Target (see page 1597)	Gets or sets the destination for the alias.

UrlAliasAutoData.AliasName Property

Gets or sets the alias name for the URL.

C#

```
public String AliasName;
```

Visual Basic

```
Public Property AliasName() As String
```

Returns

The alias name for the URL.

UrlAliasAutoData.AutoAliasType Property

Gets or sets the automatic alias source type. Types are Taxonomy (see page 876) and Folder (see page 683).

C#

```
public EkEnumeration.AutoAliasType AutoAliasType;
```

Visual Basic

```
Public Property AutoAliasType() As EkEnumeration.AutoAliasType
```

Returns

The automatic alias source type.

UrlAliasAutoData.AutoId Property

Gets the unique ID generated for this record.

C#

```
public Long AutoId;
```

Visual Basic

```
Public Property AutoId() As Long
```

Returns

The unique ID generated for this record.

UrlAliasAutoData.Example Property

Gets or sets a sample of the selected configuration.

C#

```
public String Example;
```

Visual Basic

```
Public Property Example() As String
```

Returns

A sample of the selected configuration.

UrlAliasAutoData.ExcludedPath Property

Returns the folder or taxonomy path excluded from the selected path.

C#

```
public String ExcludedPath;
```

Visual Basic

```
Public Property ExcludedPath() As String
```

UrlAliasAutoData.FileExtension Property

Gets or sets the selected file extension.

C#

```
public String FileExtension;
```

Visual Basic

```
Public Property FileExtension() As String
```

Returns

The selected file extension.

UrlAliasAutoData.FormattedSelectedPath Property

Returns the folder or taxonomy path with modifications

C#

```
public ReadOnly String FormattedSelectedPath;
```

Visual Basic

```
Public ReadOnly Property FormattedSelectedPath() As String
```

UrlAliasAutoData.HostName Property

Gets the name of the host where the content resides.

C#

```
public String HostName;
```

Visual Basic

```
Public Property HostName() As String
```

Returns

The name of the host where the content resides.

UrlAliasAutoData.IsEnabled Property

Gets or sets whether the configuration is active or not.

C#

```
public Boolean IsEnabled;
```

Visual Basic

```
Public Property IsEnabled() As Boolean
```

Returns

True if the configuration is active or not. Otherwise, it returns false.

UrlAliasAutoData.LanguageId Property

Gets the language ID for the automatic alias.

C#

```
public int LanguageId;
```

Visual Basic

```
Public Property LanguageId() As Integer
```

Returns

The language ID for the automatic alias.

UrlAliasAutoData.PageNameType Property

Gets or sets the name for the configuration. Available types are:

- ContentTitle
- ContentId
- ContentIdAndLanguage

C#

```
public EkEnumeration.AutoAliasNameType PageNameType;
```

Visual Basic

```
Public Property PageNameType() As EkEnumeration.AutoAliasNameType
```

Returns

The name for the configuration.

UrlAliasAutoData.ReplacementCharacter Property

Gets or sets the character that will replace special characters in the generated URL.

C#

```
public String ReplacementCharacter;
```

Visual Basic

```
Public Property ReplacementCharacter() As String
```

Returns

The character that will replace special characters in the generated URL.

UrlAliasAutoData.SelectedPath Property

Returns the folder or taxonomy path selected by the user

C#

```
public String SelectedPath;
```

Visual Basic

```
Public Property SelectedPath() As String
```

UrlAliasAutoData.SiteId Property

Gets or sets the root folder ID of the site. If no site is specified, returns 0 (zero).

C#

```
public Long SiteId;
```

Visual Basic

```
Public Property SiteId() As Long
```

Returns

The root folder ID of the site. If no site is specified, returns 0 (zero).

UrlAliasAutoData.SourceId Property

Gets or sets either the taxonomy ID or the folder ID depending on the auto alias type selected.

C#

```
public Long SourceId;
```

Visual Basic

```
Public Property SourceId() As Long
```

Returns

Either the taxonomy ID or the folder ID depending on the auto alias type selected.

UrlAliasAutoData.SourceName Property

Gets the name of the taxonomy or folder selected depending on the auto alias type selected.

C#

```
public String SourceName;
```

Visual Basic

```
Public Property SourceName() As String
```

Returns

The name of the taxonomy or folder selected depending on the auto alias type selected.

UrlAliasAutoData.Target Property

Gets or sets the destination for the alias.

C#

```
public String Target;
```

Visual Basic

```
Public Property Target() As String
```

Returns

The destination for the alias.

UrlAliasInfo Class

Class Hierarchy

Ektron.Cms.Common.UrlAliasInfo

C#

```
public class UrlAliasInfo;
```

Visual Basic

```
Public Class UrlAliasInfo
```

File

UrlAliasInfo.vb

Description

This is class Ektron.Cms.Common.UrlAliasInfo.

UrlAliasInfo Properties

The properties of the UrlAliasInfo class are listed here.

Public Properties

	Name	Description
	AliasId (see page 1598)	Gets or sets the alias Id for the Content (see page 822).
	CurrentAliasExtension (see page 1598)	Gets or sets the current alias extension for the Content (see page 822).
	CurrentAliasName (see page 1598)	Gets or sets the current alias name for the Content (see page 822).
	PreviousAliasExtension (see page 1598)	Gets or sets the previous alias extension for the Content (see page 822).
	PreviousAliasName (see page 1599)	Gets or sets the previous alias name for the Content (see page 822).

UrlAliasInfo.AliasId Property

Gets or sets the alias Id for the Content (see page 822).

C#

```
public Long AliasId;
```

Visual Basic

```
Public Property AliasId() As Long
```

Returns

The alias Id for the Content (see page 822).

UrlAliasInfo.CurrentAliasExtension Property

Gets or sets the current alias extension for the Content (see page 822).

C#

```
public String CurrentAliasExtension;
```

Visual Basic

```
Public Property CurrentAliasExtension() As String
```

Returns

The current alias extension for the Content (see page 822).

UrlAliasInfo.CurrentAliasName Property

Gets or sets the current alias name for the Content (see page 822).

C#

```
public String CurrentAliasName;
```

Visual Basic

```
Public Property CurrentAliasName() As String
```

Returns

The current alias name for the Content (see page 822).

UrlAliasInfo.PreviousAliasExtension Property

Gets or sets the previous alias extension for the Content (see page 822).

C#

```
public String PreviousAliasExtension;
```

Visual Basic

```
Public Property PreviousAliasExtension() As String
```

Returns

The previous alias extension for the Content (see page 822).

UrlAliasInfo.PreviousAliasName Property

Gets or sets the previous alias name for the Content (see page 822).

C#

```
public String PreviousAliasName;
```

Visual Basic

```
Public Property PreviousAliasName() As String
```

Returns

The previous alias name for the Content (see page 822).

UrlAliasManualData Class

A data class used to get or set information about a Manual URL Alias.

Class Hierarchy

Ektron.Cms.Common.UrlAliasManualData

C#

```
public class UrlAliasManualData;
```

Visual Basic

```
Public Class UrlAliasManualData
```

File

UrlAliasManualData.vb

UrlAliasManualData Properties

The properties of the UrlAliasManualData class are listed here.

Public Properties

	Name	Description
	AliasId (see page 1600)	Gets the unique ID generated for this record.
	AliasName (see page 1600)	Gets or sets the alias name for the URL without the extension.
	AliasPageName (see page 1600)	Gets the name of the alias without the query string.
	ContentId (see page 1600)	Gets or sets the Content (see page 822) ID.
	ContentLanguage (see page 1601)	Gets the content language.
	ContentTitle (see page 1601)	Gets the title of the content.

	DisplayAlias (see page 1601)	Gets the alias URL with the extension and query parameters.
	FileExtension (see page 1601)	Gets or sets the file extension for the alias.
	HostName (see page 1602)	Gets the name of the host where the content resides.
	IsDefault (see page 1602)	Gets or sets which alias name should show when the content has multiple aliases.
	IsEnabled (see page 1602)	Gets or sets whether the alias name is active or not.
	LibraryId (see page 1602)	Gets or sets the Library (see page 695) ID for the content.
	QueryString (see page 1603)	Gets or sets the querystring parameters to append to the aliased URL.
	QueryStringAction (see page 1603)	Gets or sets the querystringaction parameters.
	SiteId (see page 1603)	Gets the root folder ID of the site. If no site is specified, returns 0 (zero).
	Target (see page 1603)	Gets or sets the destination URL for the alias.

UrlAliasManualData.AliasId Property

Gets the unique ID generated for this record.

C#

```
public Long AliasId;
```

Visual Basic

```
Public Property AliasId() As Long
```

Returns

The unique ID generated for this record.

UrlAliasManualData.AliasName Property

Gets or sets the alias name for the URL without the extension.

C#

```
public String AliasName;
```

Visual Basic

```
Public Property AliasName() As String
```

Returns

The alias name for the URL without the extension.

UrlAliasManualData.AliasPageName Property

Gets the name of the alias without the query string.

C#

```
public String AliasPageName;
```

Visual Basic

```
Public Property AliasPageName() As String
```

Returns

The name of the alias without the query string.

UrlAliasManualData.ContentId Property

Gets or sets the Content (see page 822) ID.

C#

```
public Long ContentId;
```

Visual Basic

```
Public Property ContentId() As Long
```

Returns

The content ID.

UrlAliasManualData.ContentLanguage Property

Gets the content language.

C#

```
public int ContentLanguage;
```

Visual Basic

```
Public Property ContentLanguage() As Integer
```

Returns

The content language.

UrlAliasManualData.ContentTitle Property

Gets the title of the content.

C#

```
public String ContentTitle;
```

Visual Basic

```
Public Property ContentTitle() As String
```

Returns

The title of the content.

UrlAliasManualData.DisplayAlias Property

Gets the alias URL with the extension and query parameters.

C#

```
public String DisplayAlias;
```

Visual Basic

```
Public Property DisplayAlias() As String
```

Returns

The alias URL with the extension and query parameters.

Remarks

This is a computed field.

UrlAliasManualData.FileExtension Property

Gets or sets the file extension for the alias.

C#

```
public String FileExtension;
```

Visual Basic

```
Public Property FileExtension() As String
```

Returns

The file extension for the alias.

UrlAliasManualData.HostName Property

Gets the name of the host where the content resides.

C#

```
public String HostName;
```

Visual Basic

```
Public Property HostName() As String
```

Returns

The name of the host where the content resides.

UrlAliasManualData.IsDefault Property

Gets or sets which alias name should show when the content has multiple aliases.

C#

```
public Boolean IsDefault;
```

Visual Basic

```
Public Property IsDefault() As Boolean
```

Returns

Which alias name should show when the content has multiple aliases.

UrlAliasManualData.IsEnabled Property

Gets or sets whether the alias name is active or not.

C#

```
public Boolean IsEnabled;
```

Visual Basic

```
Public Property IsEnabled() As Boolean
```

Returns

Returns true if the alias name is active. Otherwise, it returns false.

UrlAliasManualData.LibraryId Property

Gets or sets the Library (see page 695) ID for the content.

C#

```
public Long LibraryId;
```

Visual Basic

```
Public Property LibraryId() As Long
```

Returns

The Library (see page 695) ID for the content.

UrlAliasManualData.QueryString Property

Gets or sets the querystring parameters to append to the aliased URL.

C#

```
public String QueryString;
```

Visual Basic

```
Public Property QueryString() As String
```

Returns

The querystring parameters to append to the aliased URL.

UrlAliasManualData.QueryStringAction Property

Gets or sets the querystringaction parameters.

C#

```
public EkEnumeration.QueryStringActionType QueryStringAction;
```

Visual Basic

```
Public Property QueryStringAction() As EkEnumeration.QueryStringActionType
```

Returns

The querystring parameters.

UrlAliasManualData.SiteId Property

Gets the root folder ID of the site. If no site is specified, returns 0 (zero).

C#

```
public Long SiteId;
```

Visual Basic

```
Public Property SiteId() As Long
```

Returns

The root folder ID of the site. If no site is specified, returns 0 (zero).

UrlAliasManualData.Target Property

Gets or sets the destination URL for the alias.

C#

```
public String Target;
```

Visual Basic

```
Public Property Target() As String
```

Returns

The destination URL for the alias.

UrlAliasRegExData Class

A data class used to get or set information about a URL Alias that uses RegEx.

Class Hierarchy

Ektron.Cms.Common.UrlAliasRegExData

C#

```
public class UrlAliasRegExData;
```

Visual Basic

```
Public Class UrlAliasRegExData
```

File

UrlAliasRegExData.vb

UrlAliasRegExData Properties

The properties of the UrlAliasRegExData class are listed here.

Public Properties

	Name	Description
	Expression (see page 1604)	Gets or sets the regular expression to be used.
	ExpressionMap (see page 1605)	Gets or sets the destination for the expression.
	ExpressionName (see page 1605)	Gets or sets a descriptive name for the expression.
	IsEnabled (see page 1605)	Gets or sets whether the expression is active or not.
	Priority (see page 1605)	Gets or sets the look up order for the expression list. Priorities are: • None • High • Medium • Low
	RegExId (see page 1606)	Gets the unique ID generated for each record.
	SiteId (see page 1606)	Gets or sets the root folder ID of the site. Returns zero if no site is specified.
	TransformedUrl (see page 1606)	Gets or sets a sample of how the ExpressionMap (see page 1605) URL will look.

UrlAliasRegExData.Expression Property

Gets or sets the regular expression to be used.

C#

```
public String Expression;
```

Visual Basic

```
Public Property Expression() As String
```

Returns

The regular expression to be used.

UrlAliasRegExData.ExpressionMap Property

Gets or sets the destination for the expression.

C#

```
public String ExpressionMap;
```

Visual Basic

```
Public Property ExpressionMap() As String
```

Returns

The destination for the expression.

UrlAliasRegExData.ExpressionName Property

Gets or sets a descriptive name for the expression.

C#

```
public String ExpressionName;
```

Visual Basic

```
Public Property ExpressionName() As String
```

Returns

A descriptive name for the expression.

UrlAliasRegExData.IsEnabled Property

Gets or sets whether the expression is active or not.

C#

```
public Boolean IsEnabled;
```

Visual Basic

```
Public Property IsEnabled() As Boolean
```

Returns

Returns true if the expression is active. Otherwise, it returns false.

UrlAliasRegExData.Priority Property

Gets or sets the look up order for the expression list. Priorities are:

- None
- High
- Medium
- Low

C#

```
public EkEnumeration.RegExPriority Priority;
```

Visual Basic

```
Public Property Priority() As EkEnumeration.RegExPriority
```

Returns

The look up order for the expression list.

UrlAliasRegExData.RegExId Property

Gets the unique ID generated for each record.

C#

```
public Long RegExId;
```

Visual Basic

```
Public Property RegExId() As Long
```

Returns

The unique ID generated for each record.

UrlAliasRegExData.SiteId Property

Gets or sets the root folder ID of the site. Returns zero if no site is specified.

C#

```
public Long SiteId;
```

Visual Basic

```
Public Property SiteId() As Long
```

Returns

The root folder ID of the site. Returns zero if no site is specified.

UrlAliasRegExData.TransformedUrl Property

Gets or sets a sample of how the ExpressionMap (see page 1605) URL will look.

C#

```
public String TransformedUrl;
```

Visual Basic

```
Public Property TransformedUrl() As String
```

Returns

A sample of how the ExpressionMap (see page 1605) URL will look.

AssetUpdateData Class

Class Hierarchy

Ektron.Cms.AssetUpdateData

C#

```
public class AssetUpdateData;
```

Visual Basic

```
Public Class AssetUpdateData
```

File

AssetUpdateData.vb

Description

This is class Ektron.Cms.AssetUpdateData.

AssetUpdateData Properties

The properties of the AssetUpdateData class are listed here.

Public Properties

	Name	Description
	AssetId (see page 1607)	Gets or sets the Asset (see page 822) ID of the content item for the AssetInfoData object.
	Comment (see page 1607)	Gets or sets comment information associated with a content item for the ContentData object.
	ContentId (see page 1608)	Gets or sets the Content (see page 822) ID of the content item for the AssetInfoData object.
	EndDate (see page 1608)	Gets or sets the end date of a content item.
	EndDateAction (see page 1608)	Gets or sets the action to be taken on the end date of a content item.
	FileName (see page 1609)	Gets or sets the filename of the asset.
	FolderId (see page 1609)	Gets or sets the Folder (see page 683) ID of the content item for the AssetInfoData object.
	GoLive (see page 1609)	Gets or sets the start date a content item goes live on the site.
	LanguageId (see page 1609)	Get or sets the language ID of the content item for the ContentData object.
	MetaData (see page 1610)	Gets or sets the Metadata of a content item as an array of ContentMetadata. This is for the ContentData object.
	TaxonomyTreeIds (see page 1610)	Gets or sets the taxonomy IDs separated by commas.
	Teaser (see page 1610)	Gets or sets a content's summary information for the ContentData object.
	Title (see page 1610)	Gets or sets the content item's title for the ContentData object.

AssetUpdateData.AssetId Property

Gets or sets the Asset (see page 822) ID of the content item for the AssetInfoData object.

C#

```
public String AssetId;
```

Visual Basic

```
Public Property AssetId() As String
```

Returns

Asset (see page 822) ID of the content item.

AssetUpdateData.Comment Property

Gets or sets comment information associated with a content item for the ContentData object.

C#

```
public String Comment;
```

Visual Basic

```
Public Property Comment() As String
```

Returns

Comment information associated with a content item.

AssetUpdateData.ContentId Property

Gets or sets the Content (see page 822) ID of the content item for the AssetInfoData object.

C#

```
public Long ContentId;
```

Visual Basic

```
Public Property ContentId() As Long
```

Returns

Integer ID of the content item.

AssetUpdateData.EndDate Property

Gets or sets the end date of a content item.

C#

```
public String EndDate;
```

Visual Basic

```
Public Property EndDate() As String
```

Returns

A string value that represents the end date of a content item.

AssetUpdateData.EndDateAction Property

Gets or sets the action to be taken on the end date of a content item.

C#

```
public CMSEndDateAction EndDateAction;
```

Visual Basic

```
Public Property EndDateAction() As CMSEndDateAction
```

Returns

A value that represents the action on the end date of a content item.

AssetUpdateData.FileName Property

Gets or sets the filename of the asset.

C#

```
public String FileName;
```

Visual Basic

```
Public Property FileName() As String
```

Returns

Asset (see page 822) File Name.

AssetUpdateData.FolderId Property

Gets or sets the Folder (see page 683) ID of the content item for the AssetInfoData object.

C#

```
public Long FolderId;
```

Visual Basic

```
Public Property FolderId() As Long
```

Returns

Folder (see page 683) ID of the content item.

AssetUpdateData.GoLive Property

Gets or sets the start date a content item goes live on the site.

C#

```
public String GoLive;
```

Visual Basic

```
Public Property GoLive() As String
```

Returns

A string value that represents the date a content item goes live.

AssetUpdateData.LanguageId Property

Get or sets the language ID of the content item for the ContentData object.

C#

```
public int LanguageId;
```

Visual Basic

```
Public Property LanguageId() As Integer
```

Returns

Language ID as an integer. For example, 1033 (English).

AssetUpdateData.MetaData Property

Gets or sets the Metadata of a content item as an array of ContentMetadata. This is for the ContentData object.

C#

```
public AssetUpdateMetaData() MetaData;
```

Visual Basic

```
Public Property MetaData() As AssetUpdateMetaData()
```

Returns

Metadata as an array of AssetMetaData.

AssetUpdateData.TaxonomyTreeIds Property

Gets or sets the taxonomy IDs separated by commas.

C#

```
public String TaxonomyTreeIds;
```

Visual Basic

```
Public Property TaxonomyTreeIds() As String
```

Returns

Asset (see page 822) File Name.

AssetUpdateData.Teaser Property

Gets or sets a content's summary information for the ContentData object.

C#

```
public String Teaser;
```

Visual Basic

```
Public Property Teaser() As String
```

Returns

Summary information for the content.

AssetUpdateData.Title Property

Gets or sets the content item's title for the ContentData object.

C#

```
public String Title;
```

Visual Basic

```
Public Property Title() As String
```

Returns

Title of the content item.

ContentRatingLabels Class

Provides a convenient data class for getting and setting Content (see page 822) Rating Labels.

Class Hierarchy

Ektron.Cms.ContentRatingLabels

C#

```
public class ContentRatingLabels;
```

Visual Basic

```
Public Class ContentRatingLabels
```

File

ContentRatingLabels.vb

ContentRatingLabels Properties

The properties of the ContentRatingLabels class are listed here.

Public Properties

	Name	Description
	AlreadyRatedMessage (see page 1611)	The message displayed to a user when they rate content, but have already rated it. Defaults to "You have already rated this content."
	BadLabel (see page 1612)	The label corresponding to a "Bad" content rating. This defaults to "Poor".
	CharsRemainingLabel (see page 1612)	The label indicating to the user how many more characters they may enter into the 'comments' section of the rating form. Defaults to "Characters Remaining: ".
	CookiesRequiredMessage (see page 1612)	The message displayed to a user when they rate content, but they have cookies disabled. Defaults to "We're sorry, but you must enable cookies in your browser to rate this content."
	FeedbackHeader (see page 1612)	The label corresponding to how the user would rate the content block. Defaults to "How would you rate the usefulness of this content?".
	GoodLabel (see page 1613)	The label corresponding to a "Good" content rating. This defaults to "Outstanding".
	RatingLevelLabel (see page 1613)	The label used to indicate the meaning of the rating displayed to the user, e.g. 7.8/9. Defaults to "Rating Level: ".
	ThankYouMessage (see page 1613)	The label displayed to a user thanking them for their content rating. Defaults to "Thank you for your feedback.".
	TotalRatingsLabel (see page 1613)	The label used to indicate the meaning of the total ratings displayed to the user, e.g. "Total Ratings: 986". Defaults to "Total Ratings: "
	UserCommentsHeader (see page 1613)	The label corresponding to why the user rated content the way they did. Defaults to "Tell us why you rated the content this way".
	ValidationMessage (see page 1614)	The label displayed to a user when they try to submit a content rating without selecting a value of 1-9. Defaults to "Please select a rating before submitting your feedback.".

ContentRatingLabels.AlreadyRatedMessage Property

The message displayed to a user when they rate content, but have already rated it. Defaults to "You have already rated this content."

C#

```
public String AlreadyRatedMessage;
```

Visual Basic

```
Public Property AlreadyRatedMessage() As String
```

ContentRatingLabels.BadLabel Property

The label corresponding to a "Bad" content rating. This defaults to "Poor".

C#

```
public String BadLabel;
```

Visual Basic

```
Public Property BadLabel() As String
```

ContentRatingLabels.CharsRemainingLabel Property

The label indicating to the user how many more characters they may enter into the 'comments' section of the rating form. Defaults to "Characters Remaining: ".

C#

```
public String CharsRemainingLabel;
```

Visual Basic

```
Public Property CharsRemainingLabel() As String
```

ContentRatingLabels.CookiesRequiredMessage Property

The message displayed to a user when they rate content, but they have cookies disabled. Defaults to "We're sorry, but you must enable cookies in your browser to rate this content."

C#

```
public String CookiesRequiredMessage;
```

Visual Basic

```
Public Property CookiesRequiredMessage() As String
```

ContentRatingLabels.FeedbackHeader Property

The label corresponding to how the user would rate the content block. Defaults to "How would you rate the usefulness of this content?".

C#

```
public String FeedbackHeader;
```

Visual Basic

```
Public Property FeedbackHeader() As String
```

ContentRatingLabels.GoodLabel Property

The label corresponding to a "Good" content rating. This defaults to "Outstanding".

C#

```
public String GoodLabel;
```

Visual Basic

```
Public Property GoodLabel() As String
```

ContentRatingLabels.RatingLevelLabel Property

The label used to indicate the meaning of the rating displayed to the user, e.g. 7.8/9. Defaults to "Rating Level: ".

C#

```
public String RatingLevelLabel;
```

Visual Basic

```
Public Property RatingLevelLabel() As String
```

ContentRatingLabels.ThankYouMessage Property

The label displayed to a user thanking them for their content rating. Defaults to "Thank you for your feedback.".

C#

```
public String ThankYouMessage;
```

Visual Basic

```
Public Property ThankYouMessage() As String
```

ContentRatingLabels.TotalRatingsLabel Property

The label used to indicate the meaning of the total ratings displayed to the user, e.g. "Total Ratings: 986". Defaults to "Total Ratings: "

C#

```
public String TotalRatingsLabel;
```

Visual Basic

```
Public Property TotalRatingsLabel() As String
```

ContentRatingLabels.UserCommentsHeader Property

The label corresponding to why the user rated content the way they did. Defaults to "Tell us why you rated the content this way".

C#

```
public String UserCommentsHeader;
```

Visual Basic

```
Public Property UserCommentsHeader() As String
```

ContentRatingLabels.ValidationMessage Property

The label displayed to a user when they try to submit a content rating without selecting a value of 1-9. Defaults to "Please select a rating before submitting your feedback.".

C#

```
public String ValidationMessage;
```

Visual Basic

```
Public Property ValidationMessage() As String
```

DiscussionCategory Class

Class Hierarchy

Ektron.Cms.DiscussionCategory

C#

```
public class DiscussionCategory;
```

Visual Basic

```
Public Class DiscussionCategory
```

File

DiscussionCategory.vb

Description

This is class Ektron.Cms.DiscussionCategory.

DiscussionCategory Properties

The properties of the DiscussionCategory class are listed here.

Public Properties

	Name	Description
	BoardID (see page 1614)	Gets or sets a Discussion Board's ID.
	CategoryID (see page 1615)	Gets or sets a Taxonomy (see page 876) category ID for a Discussion Board.
	Name (see page 1615)	Gets or sets the name of the Discussion Board.
R	SortOrder (see page 1615)	Gets the sort order of the Discussion Board.

DiscussionCategory.BoardID Property

Gets or sets a Discussion Board's ID.

C#

```
public Long BoardID;
```

Visual Basic

```
Public Property BoardID() As Long
```

DiscussionCategory.CategoryID Property

Gets or sets a Taxonomy (see page 876) category ID for a Discussion Board.

C#

```
public Long CategoryID;
```

Visual Basic

```
Public Property CategoryID() As Long
```

DiscussionCategory.Name Property

Gets or sets the name of the Discussion Board.

C#

```
public String Name;
```

Visual Basic

```
Public Property Name() As String
```

DiscussionCategory.SortOrder Property

Gets the sort order of the Discussion Board.

C#

```
public ReadOnly int SortOrder;
```

Visual Basic

```
Public ReadOnly Property SortOrder() As Integer
```

FolderRequest Class

Class Hierarchy

Ektron.Cms.FolderRequest

C#

```
public class FolderRequest;
```

Visual Basic

```
Public Class FolderRequest
```

File

FolderData.vb

Description

This is class Ektron.Cms.FolderRequest.

FolderRequest Properties

The properties of the FolderRequest class are listed here.

Public Properties

	Name	Description
	BreakInheritButton (see page 1617)	Gets or sets whether to break the folder's permission inheritance for the FolderRequest (see page 1615) data object. **True** = break a folder's permission inheritance. **False** = Do not break a folder's permission inheritance.
	CategoryRequired (see page 1618)	Flag (see page 1231) to indicate if a category selection is required while adding/editing content. This is for the FolderRequest (see page 1615) data object.
	ContentSubAssignments (see page 1618)	Gets or sets the folder's content subscription assignments. This is for the FolderRequest (see page 1615) data object.
	DomainProduction (see page 1618)	Gets or sets the production domain for the folder when the folder is a domain folder and the site is a production site. This is for the FolderRequest (see page 1615) data object.
	DomainStaging (see page 1618)	Gets or sets the staging domain for the folder when the folder is a domain folder and the site is a staging site. This is for the FolderRequest (see page 1615) data object.
	EnableReplication (see page 1619)	Gets or sets whether the folder is replicated during Quick Deploy. This is for the FolderRequest (see page 1615) data object. **True** = replicate the folder during Quick Deploy. **False** = Do not replicate the folder during Quick Deploy.
	FileDirectory (see page 1619)	Gets or sets the physical path to the folder's file directory for the FolderRequest (see page 1615) data object.
	FlagDefIdList (see page 1619)	Gets or sets a list of flag definition IDs values separated by comma. This is for the FolderRequest (see page 1615) data object.
	FlagInherited (see page 1620)	Set the property to true to inherit flag from a folder. The folder ID must be specified in flagInheritedFrom property. This is for the FolderRequest (see page 1615) data object. **1** = this folder inherits flag from a folder. **0** = this folder does not inherit flag from a folder.
	FlagInheritedFrom (see page 1620)	Gets or sets the numeric folder ID from which flag is inherited. This is for the FolderRequest (see page 1615) data object.
	FolderCfldAssignments (see page 1620)	Gets or set a list of metadata assigned to the folder. This is for the FolderRequest (see page 1615) data object.
	FolderDescription (see page 1621)	Gets or sets the folder description for the FolderRequest (see page 1615) data object.
	FolderId (see page 1621)	Gets or sets the FolderId for the FolderRequest (see page 1615) data object.
	FolderName (see page 1621)	Gets or sets the FolderName for the FolderRequest (see page 1615) data object.
	FolderType (see page 1621)	Gets or sets the folder type of the folder. This enumeration is available at Ektron.Cms.Common.EkEnumeration.FolderType. Available Types: • **0** = Content (see page 822) - a content folder that contains assets, documents and HTML content. • **1** = Blog (see page 826) - designates the folder for use with the Blogging feature. • **2** = Domain - designates the folder as a domain folder when using the Multi-site feature. • **3** = Discussion Board - designates the folder for use as a Threaded Discussion folder. • **4** = Discussion Forum (see page 264) - a subfolder of the Threaded Discussion folder. • **5** = Root - Specifies a folder as root folder. This option is... more (see page 1621)
	ImageDirectory (see page 1622)	Gets or sets the physical path to the folder's image directory for the FolderRequest (see page 1615) data object.

	IsDomainFolder (see page 1622)	Gets or sets whether the folder is a domain folder for the FolderRequest (see page 1615) data object. **True** = designate a folder as a domain folder. **False** = folder is not a domain folder.
	MetaInherited (see page 1622)	Gets or sets whether the folder inherits metadata from another folder. The folder ID must be specified in MetaInheritedFrom (see page 1623) property. • **1** = inherits metadata from a folder. • **0** = does not inherit metadata from a folder.
	MetaInheritedFrom (see page 1623)	Gets or Sets the folder ID from which metadata is inherited. This is for the FolderRequest (see page 1615) data object.
	ParentId (see page 1623)	Gets or sets the folder's parent folder ID for the FolderRequest (see page 1615) data object.
	PublishActive (see page 1623)	Gets or sets whether assets in the folder are published in another format. It also determines in which format the assets are published. For example, HTML or PDF. This is for the FolderRequest (see page 1615) data object. • **PublishPdfActive** = publish asset as a PDF. • **PublishHtmlActive** = publish asset as HTML. • **""** (Empty) - assets in the folder are not published.
	SiteMapPath (see page 1624)	Gets or sets a folder's SiteMap (see page 732) path for the FolderRequest (see page 1615) data object.
	SiteMapPathInherit (see page 1624)	Gets or sets whether the folder's SiteMap (see page 732) path is inherited for the FolderRequest (see page 1615) data object. **True** = SiteMap (see page 732) path is inherited. **False** = SiteMap (see page 732) path is not inherited.
	StyleSheet (see page 1624)	Gets or sets the folder's style sheet for the FolderRequest (see page 1615) data object.
	SubscriptionProperties (see page 1625)	Gets or sets the folder's Subscriptions properties for the FolderRequest (see page 1615) data object.
	suppressNotification (see page 1625)	Gets or sets whether to disable Web alerts. This is for the FolderRequest (see page 1615) data object. **True** = suppress Web Alert notifications. **False** = allow Web Alert notifications.
	TaxonomyIdList (see page 1625)	Gets or sets a list of taxonomy IDs values separated by comma. This is for the FolderRequest (see page 1615) data object.
	TaxonomyInherited (see page 1626)	Set the property to true to inherit taxonomy from a folder. The folder ID must be specified in TaxonomyInheritedFrom (see page 1626) property. This is for the FolderRequest (see page 1615) data object. **1** = this folder inherits taxonomy from a folder. **0** = this folder does not inherit taxonomy from a folder.
	TaxonomyInheritedFrom (see page 1626)	Gets or sets the numeric folder ID from which taxonomy is inherited. This is for the FolderRequest (see page 1615) data object.
	TemplateFileName (see page 1626)	Gets or sets the folder's template file for the FolderRequest (see page 1615) data object.
	XmlConfiguration (see page 1627)	Gets or sets the XML configuration for the folder. This is for the FolderRequest (see page 1615) data object.
	XmlInherited (see page 1627)	Gets or sets whether the XML configuration for the folder is inherited from the parent folder. This is for the FolderRequest (see page 1615) data object. **True** = inherit the XML configuration from the parent folder. **False** = Do not inherit the XML configuration from the parent folder.

FolderRequest.BreakInheritButton Property

Gets or sets whether to break the folder's permission inheritance for the FolderRequest (see page 1615) data object.

True = break a folder's permission inheritance.

False = Do not break a folder's permission inheritance.

C#

```
public Boolean BreakInheritButton;
```

Visual Basic

```
Public Property BreakInheritButton() As Boolean
```

Returns

Returns True if the folder's permission inheritance is broken.

FolderRequest.CategoryRequired Property

Flag (see page 1231) to indicate if a category selection is required while adding/editing content. This is for the FolderRequest (see page 1615) data object.

C#

```
public Boolean CategoryRequired;
```

Visual Basic

```
Public Property CategoryRequired() As Boolean
```

Returns

Boolean value. If true, a category selection is required. Otherwise, false

FolderRequest.ContentSubAssignments Property

Gets or sets the folder's content subscription assignments. This is for the FolderRequest (see page 1615) data object.

C#

```
public String ContentSubAssignments;
```

Visual Basic

```
Public Property ContentSubAssignments() As String
```

FolderRequest.DomainProduction Property

Gets or sets the production domain for the folder when the folder is a domain folder and the site is a production site. This is for the FolderRequest (see page 1615) data object.

C#

```
public String DomainProduction;
```

Visual Basic

```
Public Property DomainProduction() As String
```

Returns

The folder's existing production site domain.

Remarks

Example: www.ektron.com

FolderRequest.DomainStaging Property

Gets or sets the staging domain for the folder when the folder is a domain folder and the site is a staging site. This is for the

FolderRequest (see page 1615) data object.

C#

```
public String DomainStaging;
```

Visual Basic

```
Public Property DomainStaging() As String
```

Returns

The folder's existing staging site domain.

Remarks

Example: dev.ektron.com

FolderRequest.EnableReplication Property

Gets or sets whether the folder is replicated during Quick Deploy. This is for the FolderRequest (see page 1615) data object.

True = replicate the folder during Quick Deploy.

False = Do not replicate the folder during Quick Deploy.

C#

```
public Boolean EnableReplication;
```

Visual Basic

```
Public Property EnableReplication() As Boolean
```

Returns

Returns True if the folder is replicated during Quick Deploy.

Remarks

This property is used with the Quick Deploy feature in CMS400.NET. It does not affect Site (see page 725) Replication.

FolderRequest.FileDirectory Property

Gets or sets the physical path to the folder's file directory for the FolderRequest (see page 1615) data object.

C#

```
public String FileDirectory;
```

Visual Basic

```
Public Property FileDirectory() As String
```

Returns

An existing folder's file directory.

FolderRequest.FlagDefIdList Property

Gets or sets a list of flag definition IDs values separated by comma. This is for the FolderRequest (see page 1615) data object.

C#

```
public String FlagDefIdList;
```

Visual Basic

```
Public Property FlagDefIdList() As String
```

Returns

Comma separated list of flag definition IDs.

FolderRequest.FlagInherited Property

Set the property to true to inherit flag from a folder. The folder ID must be specified in flagInheritedFrom property. This is for the FolderRequest (see page 1615) data object.

1 = this folder inherits flag from a folder.

0 = this folder does not inherit flag from a folder.

C#

```
public Boolean FlagInherited;
```

Visual Basic

```
Public Property FlagInherited() As Boolean
```

Returns

An integer value representing whether the folder inherits flag from a folder.

Remarks

If set to true, the folder ID must be specified in flagInheritedFrom property.

FolderRequest.FlagInheritedFrom Property

Gets or sets the numeric folder ID from which flag is inherited. This is for the FolderRequest (see page 1615) data object.

C#

```
public Long FlagInheritedFrom;
```

Visual Basic

```
Public Property FlagInheritedFrom() As Long
```

Returns

An integer representing the folder ID from which flag is inherited.

Remarks

The flagInherited property must be set to 1 or True, otherwise it will be ignored.

FolderRequest.FolderCfldAssignments Property

Gets or set a list of metadata assigned to the folder. This is for the FolderRequest (see page 1615) data object.

C#

```
public String FolderCfldAssignments;
```

Visual Basic

```
Public Property FolderCfldAssignments() As String
```

FolderRequest.FolderDescription Property

Gets or sets the folder description for the FolderRequest (see page 1615) data object.

C#

```
public String FolderDescription;
```

Visual Basic

```
Public Property FolderDescription() As String
```

Returns

An existing folder's description.

Remarks

Required Field

FolderRequest.FolderId Property

Gets or sets the FolderId for the FolderRequest (see page 1615) data object.

C#

```
public Long FolderId;
```

Visual Basic

```
Public Property FolderId() As Long
```

Returns

An existing folder's integer ID.

FolderRequest.FolderName Property

Gets or sets the FolderName for the FolderRequest (see page 1615) data object.

C#

```
public String FolderName;
```

Visual Basic

```
Public Property FolderName() As String
```

Returns

An existing folder's name.

Remarks

Required Field

FolderRequest.FolderType Property

Gets or sets the folder type of the folder. This enumeration is available at Ektron.Cms.Common.EkEnumeration.FolderType.

Available Types:

- **0** = Content (see page 822) - a content folder that contains assets, documents and HTML content.
- **1** = Blog (see page 826) - designates the folder for use with the Blogging feature.
- **2** = Domain - designates the folder as a domain folder when using the Multi-site feature.
- **3** = Discussion Board - designates the folder for use as a Threaded Discussion folder.
- **4** = Discussion Forum (see page 264) - a subfolder of the Threaded Discussion folder.
- **5** = Root - Specifies a folder as root folder. This option is not available.
- **6** = Community (see page 750) - Membership (see page 360) users can add and edit content contained in these folders.

C#

```
public int FolderType;
```

Visual Basic

```
Public Property FolderType() As Integer
```

Returns

The integer value of the folder type.

FolderRequest.ImageDirectory Property

Gets or sets the physical path to the folder's image directory for the FolderRequest (see page 1615) data object.

C#

```
public String ImageDirectory;
```

Visual Basic

```
Public Property ImageDirectory() As String
```

Returns

An existing folder's image directory.

FolderRequest.IsDomainFolder Property

Gets or sets whether the folder is a domain folder for the FolderRequest (see page 1615) data object.

True = designate a folder as a domain folder.

False = folder is not a domain folder.

C#

```
public Boolean IsDomainFolder;
```

Visual Basic

```
Public Property IsDomainFolder() As Boolean
```

Returns

Returns True if the folder is a domain folder.

FolderRequest.MetaInherited Property

Gets or sets whether the folder inherits metadata from another folder. The folder ID must be specified in MetaInheritedFrom (

see page 1623) property.

- **1** = inherits metadata from a folder.
- **0** = does not inherit metadata from a folder.

C#

```
public int MetaInherited;
```

Visual Basic

```
Public Property MetaInherited() As Integer
```

Returns

An integer value representing whether the folder inherits metadata from another folder.

Remarks

If set to true, the folder ID must be specified in MetaInheritedFrom (see page 1623) property.

FolderRequest.MetaInheritedFrom Property

Gets or Sets the folder ID from which metadata is inherited. This is for the FolderRequest (see page 1615) data object.

C#

```
public Long MetaInheritedFrom;
```

Visual Basic

```
Public Property MetaInheritedFrom() As Long
```

Returns

An integer value representing the folder ID

Remarks

InheritMetadata must be set to 1, otherwise it will be ignored.

FolderRequest.ParentId Property

Gets or sets the folder's parent folder ID for the FolderRequest (see page 1615) data object.

C#

```
public Long ParentId;
```

Visual Basic

```
Public Property ParentId() As Long
```

Returns

An existing folder's parent folder ID.

FolderRequest.PublishActive Property

Gets or sets whether assets in the folder are published in another format. It also determines in which format the assets are published. For example, HTML or PDF. This is for the FolderRequest (see page 1615) data object.

- **PublishPdfActive** = publish asset as a PDF.

- **PublishHtmlActive** = publish asset as HTML.
- **""** (Empty) - assets in the folder are not published.

C#

```
public String PublishActive;
```

Visual Basic

```
Public Property PublishActive() As String
```

Returns

A string based on whether to publish contents as HTML, PDF or not at all.

FolderRequest.SiteMapPath Property

Gets or sets a folder's SiteMap (see page 732) path for the FolderRequest (see page 1615) data object.

C#

```
public Ektron.Cms.Common.SitemapPath() SiteMapPath;
```

Visual Basic

```
Public Property SiteMapPath() As Ektron.Cms.Common.SitemapPath()
```

Returns

An existing folder's SiteMap (see page 732) path.

FolderRequest.SiteMapPathInherit Property

Gets or sets whether the folder's SiteMap (see page 732) path is inherited for the FolderRequest (see page 1615) data object.

True = SiteMap (see page 732) path is inherited.

False = SiteMap (see page 732) path is not inherited.

C#

```
public Boolean SiteMapPathInherit;
```

Visual Basic

```
Public Property SiteMapPathInherit() As Boolean
```

Returns

True or False depending whether an existing folder's SiteMap (see page 732) path is inherited or not.

FolderRequest.StyleSheet Property

Gets or sets the folder's style sheet for the FolderRequest (see page 1615) data object.

C#

```
public String StyleSheet;
```

Visual Basic

```
Public Property StyleSheet() As String
```

Returns

An existing folder's style sheet name.

FolderRequest.SubscriptionProperties Property

Gets or sets the folder's Subscriptions properties for the FolderRequest (see page 1615) data object.

C#

```
public SubscriptionPropertiesData SubscriptionProperties;
```

Visual Basic

```
Public Property SubscriptionProperties() As SubscriptionPropertiesData
```

FolderRequest.suppressNotification Property

Gets or sets whether to disable Web alerts. This is for the FolderRequest (see page 1615) data object.

True = suppress Web Alert notifications.

False = allow Web Alert notifications.

C#

```
public Boolean suppressNotification;
```

Visual Basic

```
Public Property suppressNotification() As Boolean
```

Description

True = suppress Web Alert notifications.

False = allow Web Alert notifications.

Default = False, to override this use the BreakInheritance property of the FolderRequest.SubscriptionProperties (see page 1625) object (set it to True).

Remarks

Boolean value. If true, suppress Web Alert notifications. Otherwise, false.

FolderRequest.TaxonomyIdList Property

Gets or sets a list of taxonomy IDs values separated by comma. This is for the FolderRequest (see page 1615) data object.

C#

```
public String TaxonomyIdList;
```

Visual Basic

```
Public Property TaxonomyIdList() As String
```

Returns

Comma separated list of taxonomy IDs.

FolderRequest.TaxonomyInherited Property

Set the property to true to inherit taxonomy from a folder. The folder ID must be specified in TaxonomyInheritedFrom (see page 1626) property. This is for the FolderRequest (see page 1615) data object.

1 = this folder inherits taxonomy from a folder.

0 = this folder does not inherit taxonomy from a folder.

C#

```
public Boolean TaxonomyInherited;
```

Visual Basic

```
Public Property TaxonomyInherited() As Boolean
```

Returns

An integer value representing whether the folder inherits taxonomy from a folder.

Remarks

If set to true, the folder ID must be specified in TaxonomyInheritedFrom (see page 1626) property.

FolderRequest.TaxonomyInheritedFrom Property

Gets or sets the numeric folder ID from which taxonomy is inherited. This is for the FolderRequest (see page 1615) data object.

C#

```
public Long TaxonomyInheritedFrom;
```

Visual Basic

```
Public Property TaxonomyInheritedFrom() As Long
```

Returns

An integer representing the folder ID from which taxonomy is inherited.

Remarks

The TaxonomyInherited (see page 1626) property must be set to 1 or True, otherwise it will be ignored.

FolderRequest.TemplateFileName Property

Gets or sets the folder's template file for the FolderRequest (see page 1615) data object.

C#

```
public String TemplateFileName;
```

Visual Basic

```
Public Property TemplateFileName() As String
```

Returns

An existing folder's template name.

FolderRequest.XmlConfiguration Property

Gets or sets the XML configuration for the folder. This is for the FolderRequest (see page 1615) data object.

C#

```
public String XmlConfiguration;
```

Visual Basic

```
Public Property XmlConfiguration() As String
```

Returns

The existing XML configuration for the folder.

FolderRequest.XmlInherited Property

Gets or sets whether the XML configuration for the folder is inherited from the parent folder. This is for the FolderRequest (see page 1615) data object.

True = inherit the XML configuration from the parent folder.

False = Do not inherit the XML configuration from the parent folder.

C#

```
public Boolean XmlInherited;
```

Visual Basic

```
Public Property XmlInherited() As Boolean
```

Returns

Returns True if the XML configuration is inherited from the parent folder.

FormFieldDataItem Class

Describes an instance of a data value from an individual form field. The data is a name/value pair.

Class Hierarchy

Ektron.Cms.FormFieldDataItem

C#

```
public class FormFieldDataItem;
```

Visual Basic

```
Public Class FormFieldDataItem
```

File

FormFieldDataItem.vb

Remarks

Typically, this class is used in an array.

FormFieldDataItem Properties

The properties of the FormFieldDataItem class are listed here.

Public Properties

	Name	Description
	DataValue (see page 1628)	Gets or sets the value of the form field.
	FieldName (see page 1628)	Gets or sets the name of the form field.

FormFieldDataItem.DataValue Property

Gets or sets the value of the form field.

C#

```
public Object DataValue;
```

Visual Basic

```
Public Property DataValue() As Object
```

Remarks

The data type of the value is variant (i.e., an Object).

FormFieldDataItem.FieldName Property

Gets or sets the name of the form field.

C#

```
public String FieldName;
```

Visual Basic

```
Public Property FieldName() As String
```

FormFieldDatalist Class

Describes an individual datalist suitable as an enumeration for a select list.

Class Hierarchy

Ektron.Cms.FormFieldDatalist

C#

```
public class FormFieldDatalist;
```

Visual Basic

```
Public Class FormFieldDatalist
```

File

FormFieldDatalist.vb

Remarks

Typically used as an array in class FormFieldList (see page 1636).

Typically referenced by name in class FormFieldDefinition (see page 1633).

FormFieldDatalist Properties

The properties of the FormFieldDatalist class are listed here.

Public Properties

	Name	Description
	CaptionXPath (see page 1629)	Gets or sets the CaptionXPath of the pre-defined datalist.
	DatalistItems (see page 1629)	Gets or sets the array of items defined in the datalist.
	DataNamespaces (see page 1629)	Gets or sets the Data Namespace of the pre-defined datalist.
	DataSelect (see page 1630)	Gets or sets the Data Select of the pre-defined datalist.
	DataSrc (see page 1630)	Gets or sets the Data Source of the pre-defined datalist.
	Language (see page 1630)	Gets or sets the language code of the datalist. For example, en or en-US.
	Name (see page 1631)	Gets or sets the name of the datalist.
	ValueXPath (see page 1631)	Gets or sets the Value XPath of the pre-defined datalist.

FormFieldDatalist.CaptionXPath Property

Gets or sets the CaptionXPath of the pre-defined datalist.

C#

```
[XmlAttributeAttribute("ektdesignns_captionxpath")]
public String CaptionXPath;
```

Visual Basic

```
<XmlAttributeAttribute("ektdesignns_captionxpath")>
Public Property CaptionXPath() As String
```

FormFieldDatalist.DatalistItems Property

Gets or sets the array of items defined in the datalist.

C#

```
[XmlElement("item")]
public FormFieldDatalistItem() DatalistItems;
```

Visual Basic

```
<XmlElement("item")>
Public Property DatalistItems() As FormFieldDatalistItem()
```

FormFieldDatalist.DataNamespaces Property

Gets or sets the Data Namespace of the pre-defined datalist.

C#

```
[XmlAttributeAttribute("ektdesignns_datanamespaces")]
public String DataNamespaces;
```

Visual Basic

```
<XmlAttributeAttribute("ektdesignns_datanamespaces")>
Public Property DataNamespaces() As String
```

FormFieldDatalist.DataSelect Property

Gets or sets the Data Select of the pre-defined datalist.

C#

```
[XmlAttributeAttribute("ektdesignns_dataselect")]
public String DataSelect;
```

Visual Basic

```
<XmlAttributeAttribute("ektdesignns_dataselect")>
Public Property DataSelect() As String
```

FormFieldDatalist.DataSrc Property

Gets or sets the Data Source of the pre-defined datalist.

C#

```
[XmlAttributeAttribute("ektdesignns_datasrc")]
public String DataSrc;
```

Visual Basic

```
<XmlAttributeAttribute("ektdesignns_datasrc")>
Public Property DataSrc() As String
```

FormFieldDatalist.Language Property

Gets or sets the language code of the datalist. For example, en or en-US.

C#

```
[XmlAttributeAttribute("xml:lang")]
public String Language;
```

Visual Basic

```
<XmlAttributeAttribute("xml:lang")>
Public Property Language() As String
```

Returns

Language code.

Remarks

The first two letters of the language code conform to ISO 639. The last two letters are an optional country code that conforms to ISO 3166.

FormFieldDatalist.Name Property

Gets or sets the name of the datalist.

C#

```
[XmlAttributeAttribute("name")]
public String Name;
```

Visual Basic

```
<XmlAttributeAttribute("name")>
Public Property Name() As String
```

FormFieldDatalist.ValueXPath Property

Gets or sets the Value XPath of the pre-defined datalist.

C#

```
[XmlAttributeAttribute("ektdesignns_valuexpath")]
public String ValueXPath;
```

Visual Basic

```
<XmlAttributeAttribute("ektdesignns_valuexpath")>
Public Property ValueXPath() As String
```

FormFieldDatalistItem Class

Describes an individual datalist item (most importantly its value and display value).

Class Hierarchy

Ektron.Cms.FormFieldDatalistItem

C#

```
public class FormFieldDatalistItem;
```

Visual Basic

```
Public Class FormFieldDatalistItem
```

File

FormFieldDatalistItem.vb

Remarks

Typically used as an array in class FormFieldDatalist (see page 1628) Cms.FormFieldDatalist (see page 1628)

FormFieldDatalistItem Properties

The properties of the FormFieldDatalistItem class are listed here.

Public Properties

	Name	Description
	DataValue (see page 1632)	Gets or sets the value for this item in the data list.
	DisplayValue (see page 1632)	Gets or sets the value to be displayed.
	IsDefault (see page 1632)	Gets or sets whether this item is the default value in the datalist. Set to **True** if this item is a default value for the list.
	IsEnabled (see page 1633)	Gets or sets whether this item is enabled for selection. Set to **True** if this item is enabled.

FormFieldDatalistItem.DataValue Property

Gets or sets the value for this item in the data list.

C#

```
[XmlAttributeAttribute("value")]
public String DataValue;
```

Visual Basic

```
<XmlAttributeAttribute("value")>
Public Property DataValue() As String
```

Remarks

This value is typically language independent and suitable for machine processing.

FormFieldDatalistItem.DisplayValue Property

Gets or sets the value to be displayed.

C#

```
[XmlText()]
public String DisplayValue;
```

Visual Basic

```
<XmlText()>
Public Property DisplayValue() As String
```

Remarks

This value is typically dependent on the language and suitable for human reading.

FormFieldDatalistItem.IsDefault Property

Gets or sets whether this item is the default value in the datalist. Set to **True** if this item is a default value for the list.

C#

```
[XmlAttributeAttribute("default")]
public Boolean IsDefault;
```

Visual Basic

```
<XmlAttributeAttribute("default")>
Public Property IsDefault() As Boolean
```

Remarks

If a list allows only one item to be selected at a time, then only one item should be marked as the default. If more than one may

be selected, then it is reasonable to have more than one default.

FormFieldDatalistItem.IsEnabled Property

Gets or sets whether this item is enabled for selection. Set to **True** if this item is enabled.

C#

```
[XmlAttributeAttribute("enabled")]
public Boolean IsEnabled;
```

Visual Basic

```
<XmlAttributeAttribute("enabled")>
Public Property IsEnabled() As Boolean
```

Remarks

An item may be disabled to prevent it from being selected. Typically an item, once defined and used, would not be deleted from the list, but simply disabled. This preserves the display value of the item for backward compatibility.

FormFieldDefinition Class

Describes the definition of an individual form field.

Class Hierarchy

Ektron.Cms.FormFieldDefinition

C#

```
public class FormFieldDefinition;
```

Visual Basic

```
Public Class FormFieldDefinition
```

File

FormFieldDefinition.vb

Remarks

Typically used as an array in class FormFieldList (see page 1636).

FormFieldDefinition Properties

The properties of the FormFieldDefinition class are listed here.

Public Properties

	Name	Description
	Basetype (see page 1634)	Gets or sets the basic datatype of the field.
	ContentType (see page 1634)	Gets or sets the type of content allowed within the field. This is loosely associated with the datatype.
	DatalistName (see page 1634)	Gets or sets the name of the datalist, if one is defined for this field.

	Datatype (see page 1635)	Gets or sets the datatype of the field.
	DisplayName (see page 1635)	Gets or sets the name of the field to be displayed.
	FieldName (see page 1635)	Gets or sets the name of the field.
	Language (see page 1636)	Gets or sets the language code of the datalist. For example, en or en-US.
	XPath (see page 1636)	Gets or sets the XPath expression that locates the field.

FormFieldDefinition.Basetype Property

Gets or sets the basic datatype of the field.

C#

```
[XmlAttributeAttribute("basetype")]
public String Basetype;
```

Visual Basic

```
<XmlAttributeAttribute("basetype")>
Public Property Basetype() As String
```

Remarks

The possible values of the basic datatype vary, including, text, textbox, calendar and checkbox.

FormFieldDefinition.ContentType Property

Gets or sets the type of content allowed within the field. This is loosely associated with the datatype.

C#

```
[XmlAttributeAttribute("content")]
public String ContentType;
```

Visual Basic

```
<XmlAttributeAttribute("content")>
Public Property ContentType() As String
```

Remarks

The possible values for content vary.

FormFieldDefinition.DatalistName Property

Gets or sets the name of the datalist, if one is defined for this field.

C#

```
[XmlAttributeAttribute("datalist")]
public String DatalistName;
```

Visual Basic

```
<XmlAttributeAttribute("datalist")>
Public Property DatalistName() As String
```

Description

Name of the associated datalist.

Remarks

Applies only to fields whose values are selected from a list. The allowed values are defined in the datalist.

FormFieldDefinition.Datatype Property

Gets or sets the datatype of the field.

C#

```
[XmlAttributeAttribute("datatype")]
public String Datatype;
```

Visual Basic

```
<XmlAttributeAttribute("datatype")>
Public Property Datatype() As String
```

Remarks

The possible values of the datatype vary. Generally they conform to W3C XML Schema types such as 'string' and 'date'. Other values include 'selection' and 'choice'.

FormFieldDefinition.DisplayName Property

Gets or sets the name of the field to be displayed.

C#

```
[XmlText()]
public String DisplayName;
```

Visual Basic

```
<XmlText()>
Public Property DisplayName() As String
```

Remarks

The name is suitable for human reading.

FormFieldDefinition.FieldName Property

Gets or sets the name of the field.

C#

```
[XmlAttributeAttribute("name")]
public String FieldName;
```

Visual Basic

```
<XmlAttributeAttribute("name")>
Public Property FieldName() As String
```

Remarks

The name is suitable for machine processing, such as with an XML parser.

FormFieldDefinition.Language Property

Gets or sets the language code of the datalist. For example, en or en-US.

C#

```
[XmlAttributeAttribute("xml:lang")]
public String Language;
```

Visual Basic

```
<XmlAttributeAttribute("xml:lang")>
Public Property Language() As String
```

Remarks

The first two letters of the language code conform to ISO 639. The last two letters are an optional country code that conforms to ISO 3166.

FormFieldDefinition.XPath Property

Gets or sets the XPath expression that locates the field.

C#

```
[XmlAttributeAttribute("xpath")]
public String XPath;
```

Visual Basic

```
<XmlAttributeAttribute("xpath")>
Public Property XPath() As String
```

FormFieldList Class

Describes the definition of all the fields in a specific form.

Class Hierarchy

Ektron.Cms.FormFieldList

C#

```
[XmlRoot("fieldlist")]
public class FormFieldList;
```

Visual Basic

```
<XmlRoot("fieldlist")>
Public Class FormFieldList
```

File

FormFieldList.vb

FormFieldList Properties

The properties of the FormFieldList class are listed here.

Public Properties

	Name	Description
	Datalists (see page 1637)	Gets or sets an array of defined datalists.
	Fields (see page 1637)	Gets or sets an array of field definitions.

FormFieldList.Datalists Property

Gets or sets an array of defined datalists.

C#

```
[XmlElement("datalist")]
public FormFieldDatalist() Datalists;
```

Visual Basic

```
<XmlElement("datalist")>
Public Property Datalists() As FormFieldDatalist()
```

Returns

Ektron.Cms.FormFieldDatalist (see page 1628)

FormFieldList.Fields Property

Gets or sets an array of field definitions.

C#

```
[XmlElement("field")]
public FormFieldDefinition() Fields;
```

Visual Basic

```
<XmlElement("field")>
Public Property Fields() As FormFieldDefinition()
```

Returns

Ektron.Cms.FormFieldDefinition (see page 1633)

FormRequestData Class

A data object that contains information about a form.

Class Hierarchy

Ektron.Cms.FormRequestData

C#

```
public class FormRequestData;
```

Visual Basic

```
Public Class FormRequestData
```

File

FormRequestData.vb

FormRequestData Properties

The properties of the FormRequestData class are listed here.

Public Properties

	Name	Description
	AutoFill (see page 1638)	Indicates whether a form's fields are pre-filled when the form is displayed. • **True** - a form's fields are pre-filled when the form is displayed. • **False** - a form's fields are empty when the form is displayed.
	FieldList (see page 1639)	Gets or sets a list of the fields defined in the form.
	Id (see page 1639)	The numeric ID of the form.
	IsPoll (see page 1639)	Gets whether the form is used as poll. • **True** - this form is a poll. • **False** - this form is not a poll.
	Language (see page 1639)	Gets or sets the form's language ID. For example, 1033 for English, 1034 for Spanish or 1036 for French. When set to 0 (zero), the language is dynamic.
	PostBackData (see page 1640)	Gets or sets the data submitted from the form as a collection.
	PreFillFielddata (see page 1640)	Gets or sets the data values that are used to pre-fill a form's fields. When you display a form, this is the information pre-loaded into each field. For example, in a Country: field you might have "United States" set to pre-fill the field. This way, people from the United States don't have to type it in each time. Another example is when a user is logged in and fills out a form more that once, his name and and email address always appear in their respective fields after completing the form the first time.
	TagRequired (see page 1640)	Gets or sets whether to wrap form tags around the form element. • **True** - wrap form tags around the form element. • **False** - Do not wrap form tags around the form element.

FormRequestData.AutoFill Property

Indicates whether a form's fields are pre-filled when the form is displayed.

- **True** - a form's fields are pre-filled when the form is displayed.
- **False** - a form's fields are empty when the form is displayed.

C#

```
public Boolean AutoFill;
```

Visual Basic

```
Public Property AutoFill() As Boolean
```

FormRequestData.FieldList Property

Gets or sets a list of the fields defined in the form.

C#

```
public FormFieldList FieldList;
```

Visual Basic

```
Public Property FieldList() As FormFieldList
```

Returns

Ektron.Cms.FormFieldList (see page 1636)

FormRequestData.Id Property

The numeric ID of the form.

C#

```
public Long Id;
```

Visual Basic

```
Public Property Id() As Long
```

Returns

The numeric ID (Long) of the form.

FormRequestData.IsPoll Property

Gets whether the form is used as poll.

- **True** - this form is a poll.
- **False** - this form is not a poll.

C#

```
public Boolean IsPoll;
```

Visual Basic

```
Public Property IsPoll() As Boolean
```

Remarks

This should only be read not set.

FormRequestData.Language Property

Gets or sets the form's language ID. For example, 1033 for English, 1034 for Spanish or 1036 for French. When set to 0 (zero), the language is dynamic.

C#

```
public int Language;
```

Visual Basic

```
Public Property Language() As Integer
```

Returns

The form's language ID as an integer.

FormRequestData.PostBackData Property

Gets or sets the data submitted from the form as a collection.

C#

```
public System.Collections.Specialized.NameValueCollection PostBackData;
```

Visual Basic

```
Public Property PostBackData() As System.Collections.Specialized.NameValueCollection
```

Returns

System.Collections.Specialized.NameValueCollection

FormRequestData.PreFillFielddata Property

Gets or sets the data values that are used to pre-fill a form's fields. When you display a form, this is the information pre-loaded into each field.

For example, in a Country: field you might have "United States" set to pre-fill the field. This way, people from the United States don't have to type it in each time. Another example is when a user is logged in and fills out a form more that once, his name and and email address always appear in their respective fields after completing the form the first time.

C#

```
public FormFieldDataItem() PreFillFielddata;
```

Visual Basic

```
Public Property PreFillFielddata() As FormFieldDataItem()
```

Remarks

If the AutoFill (see page 1638) property is set to false, this property appears as empty.

FormRequestData.TagRequired Property

Gets or sets whether to wrap form tags around the form element.

- **True** - wrap form tags around the form element.
- **False** - Do not wrap form tags around the form element.

C#

```
public Boolean TagRequired;
```

Visual Basic

```
Public Property TagRequired() As Boolean
```

Remarks

Typically set to false.

PagingInfo Class

Class Hierarchy

Ektron.Cms.PagingInfo

C#

```
[Serializable()]
public class PagingInfo;
```

Visual Basic

```
<Serializable()>
Public Class PagingInfo
```

File

PagingInfo.vb

Description

This is class Ektron.Cms.PagingInfo.

PagingInfo Properties

The properties of the PagingInfo class are listed here.

Public Properties

	Name	Description
	CurrentPage (see page 1641)	Get or Set the Current page for which you want records
R	EndRow (see page 1642)	The ending row number for the page.
	RecordsPerPage (see page 1642)	Gets the records per page.
R	StartRow (see page 1642)	The Starting row number for the page.
	TotalPages (see page 1642)	Gets the total pages for this result set
	TotalRecords (see page 1642)	Gets the total number of records.

PagingInfo.CurrentPage Property

Get or Set the Current page for which you want records

C#

```
public int CurrentPage;
```

Visual Basic

```
Public Property CurrentPage() As Integer
```

PagingInfo.EndRow Property

The ending row number for the page.

C#

```
public ReadOnly int EndRow;
```

Visual Basic

```
Public ReadOnly Property EndRow() As Integer
```

Remarks

This is a calculated field.

PagingInfo.RecordsPerPage Property

Gets the records per page.

C#

```
public int RecordsPerPage;
```

Visual Basic

```
Public Property RecordsPerPage() As Integer
```

PagingInfo.StartRow Property

The Starting row number for the page.

C#

```
public ReadOnly int StartRow;
```

Visual Basic

```
Public ReadOnly Property StartRow() As Integer
```

Remarks

This is a calculated field.

PagingInfo.TotalPages Property

Gets the total pages for this result set

C#

```
public int TotalPages;
```

Visual Basic

```
Public Property TotalPages() As Integer
```

PagingInfo.TotalRecords Property

Gets the total number of records.

C#

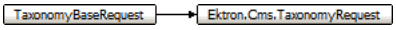
```
public int TotalRecords;
```

Visual Basic

```
Public Property TotalRecords() As Integer
```

TaxonomyRequest Class

Class Hierarchy



C#

```
[Serializable()]
public class TaxonomyRequest : TaxonomyBaseRequest;
```

Visual Basic

```
<Serializable()>
Public Class TaxonomyRequest
Inherits TaxonomyBaseRequest
```

File

TaxonomyData.vb

Description

This is class Ektron.Cms.TaxonomyRequest.

TaxonomyRequest Properties

The properties of the TaxonomyRequest class are listed here.

Public Properties

	Name	Description
	Depth (see page 1644)	Gets or sets number of taxonomy levels to retrieve below each taxonomy/category. For example, if the taxonomy is Businesses > Restaurants > Pizza , and you set this property to 2, only Business and Restaurants are available in code behind. To retrieve the all of the categories for a taxonomy recursively, enter -1. The default value is 1. **Important!** - For a live site, Ektron strongly recommends leaving this value at 1. Increasing this value can slow down your live Web server. However, for testing on a staging server, you can increase the depth.
	FolderId (see page 1644)	Deprecated. Not in use.
	IncludeItems (see page 1644)	When True, include all items from the taxonomy's structure. When false, use only the taxonomy category's structure.
	Preview (see page 1645)	Gets or sets whether preview content should be included in the taxonomy category.
	ReadCount (see page 1645)	Gets or sets the quantity of items in each taxonomy category.
	TaxonomyIdList (see page 1645)	Gets or sets a list of taxonomy items IDs associated with content.

	TaxonomyItemType (see page 1645)	Gets or sets the item type being assigned to the taxonomy. • Content (see page 822) = 0 • User (see page 950) = 1 • Group = 2 • Blogs = 3 • Forum (see page 264) = 4
	TaxonomyPath (see page 1646)	Gets or set the path of the taxonomy category. For example, "\New Hampshire\Restaurants\Pizza"

TaxonomyRequest.Depth Property

Gets or sets number of taxonomy levels to retrieve below each taxonomy/category. For example, if the taxonomy is Businesses > Restaurants > Pizza , and you set this property to 2, only Business and Restaurants are available in code behind.

To retrieve the all of the categories for a taxonomy recursively, enter -1.

The default value is 1.

Important! - For a live site, Ektron strongly recommends leaving this value at 1. Increasing this value can slow down your live Web server. However, for testing on a staging server, you can increase the depth.

C#

```
public int Depth;
```

Visual Basic

```
Public Property Depth() As Integer
```

Returns

An integer representing the depth level.

Remarks

This property should not be left empty. Enter a value at least a value of 1.

TaxonomyRequest.FolderId Property

Deprecated. Not in use.

C#

```
public Long FolderId;
```

Visual Basic

```
Public Property FolderId() As Long
```

TaxonomyRequest.IncludeItems Property

When True, include all items from the taxonomy's structure. When false, use only the taxonomy category's structure.

C#

```
public Boolean IncludeItems;
```

Visual Basic

```
Public Property IncludeItems() As Boolean
```

Returns

All items in the taxonomy's structure.

TaxonomyRequest.Preview Property

Gets or sets whether preview content should be included in the taxonomy category.

C#

```
public Boolean Preview;
```

Visual Basic

```
Public Property Preview() As Boolean
```

Returns

A boolean.

TaxonomyRequest.ReadCount Property

Gets or sets the quantity of items in each taxonomy category.

C#

```
public Boolean ReadCount;
```

Visual Basic

```
Public Property ReadCount() As Boolean
```

Returns

An integer representing the number of items.

TaxonomyRequest.TaxonomyIdList Property

Gets or sets a list of taxonomy items IDs associated with content.

C#

```
public String TaxonomyIdList;
```

Visual Basic

```
Public Property TaxonomyIdList() As String
```

Returns

A list of taxonomy items.

TaxonomyRequest.TaxonomyItemType Property

Gets or sets the item type being assigned to the taxonomy.

- Content (see page 822) = 0

- User (see page 950) = 1
- Group = 2
- Blogs = 3
- Forum (see page 264) = 4

C#

```
public TaxonomyItemType TaxonomyItemType;
```

Visual Basic

```
Public Property TaxonomyItemType() As TaxonomyItemType
```

Returns

The item's type.

TaxonomyRequest.TaxonomyPath Property

Gets or set the path of the taxonomy category. For example, "\New Hampshire\Restaurants\Pizza"

C#

```
public String TaxonomyPath;
```

Visual Basic

```
Public Property TaxonomyPath() As String
```

Returns

The path of the taxonomy category.

UriFavorite Class

A list of properties that make up a URI favorite data.

Class Hierarchy

Ektron.Cms.UriFavorite

C#

```
[Serializable()]
public class UriFavorite;
```

Visual Basic

```
<Serializable()>
Public Class UriFavorite
```

File

TaxonomyData.vb

UriFavorite Properties

The properties of the UriFavorite class are listed here.

Public Properties

	Name	Description
	FavoriteDesc (see page 1647)	Gets or sets a description for a URI favorite.
	FavoriteId (see page 1647)	Gets or sets the numeric ID of the favorite.
	FavoriteName (see page 1647)	Gets or sets the name of the URI for the favorite.
	FavoriteUri (see page 1648)	Gets or sets the actual URI value of the favorite.
	UserId (see page 1648)	Gets or set the ID of the user with which the favorite is associated.

UriFavorite.FavoriteDesc Property

Gets or sets a description for a URI favorite.

C#

```
public String FavoriteDesc;
```

Visual Basic

```
Public Property FavoriteDesc() As String
```

Returns

A description of the URI favorite.

UriFavorite.FavoriteId Property

Gets or sets the numeric ID of the favorite.

C#

```
public Long FavoriteId;
```

Visual Basic

```
Public Property FavoriteId() As Long
```

Returns

The numeric ID of a favorite.

UriFavorite.FavoriteName Property

Gets or sets the name of the URI for the favorite.

C#

```
public String FavoriteName;
```

Visual Basic

```
Public Property FavoriteName() As String
```

Returns

The name of the URI.

UriFavorite.FavoriteUri Property

Gets or sets the actual URI value of the favorite.

C#

```
public String FavoriteUri;
```

Visual Basic

```
Public Property FavoriteUri() As String
```

Returns

The URI value.

UriFavorite.UserId Property

Gets or set the ID of the user with which the favorite is associated.

C#

```
public Long UserId;
```

Visual Basic

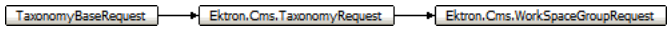
```
Public Property UserId() As Long
```

Returns

The ID of the user.

WorkSpaceGroupRequest Class

Class Hierarchy



C#

```
[Serializable()]
public class WorkSpaceGroupRequest : TaxonomyRequest;
```

Visual Basic

```
<Serializable()>
Public Class WorkSpaceGroupRequest
Inherits TaxonomyRequest
```

File

TaxonomyData.vb

Description

This is class Ektron.Cms.WorkSpaceGroupRequest.

WorkSpaceGroupRequest Properties

The properties of the WorkSpaceGroupRequest class are listed here.

Public Properties

	Name	Description
	GroupId (see page 1649)	Gets or sets the ID of the community group to associate with the taxonomy category.

WorkSpaceGroupRequest.GroupId Property

Gets or sets the ID of the community group to associate with the taxonomy category.

C#

```
public Long GroupId;
```

Visual Basic

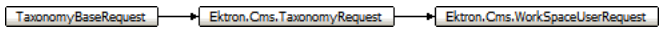
```
Public Property GroupId() As Long
```

Returns

The community group's ID.

WorkSpaceUserRequest Class

Class Hierarchy



C#

```
[Serializable()]
public class WorkSpaceUserRequest : TaxonomyRequest;
```

Visual Basic

```
<Serializable()>
Public Class WorkSpaceUserRequest
Inherits TaxonomyRequest
```

File

TaxonomyData.vb

Description

This is class Ektron.Cms.WorkSpaceUserRequest.

WorkSpaceUserRequest Properties

The properties of the WorkSpaceUserRequest class are listed here.

Public Properties

	Name	Description
	UserId (see page 1650)	Gets or sets the ID of a user to associate with the taxonomy category.

WorkSpaceUserRequest.UserId Property

Gets or sets the ID of a user to associate with the taxonomy category.

C#

```
public Long UserId;
```

Visual Basic

```
Public Property UserId() As Long
```

Returns

The user ID.

Index

.

.NET Assemblies 600

A

ActionRecipientInfo class 553
- about ActionRecipientInfo class 553
- ActionRecipientInfo 554
- ActionRecipientInfo properties 554
- DisplayName 554
- Email 554
- FirstName 555
- IsUser 555
- LastName 555
- UserId 555

ActionRequestInfo class 555
- about ActionRequestInfo class 555
- ActionRequestInfo 556
- ActionRequestInfo enumerations 556
- ActionRequestInfo properties 556
- ErrorMessage 557
- Id 557
- Recipients 557
- SenderId 557
- Type 557

ActiveDirectoryUser class 1103
- about ActiveDirectoryUser class 1103
- ActiveDirectoryUser methods 1103
- AddADGroupToCMS 1104
- DeSynchUserGroups 1104
- DeSynchUsers 1105
- GetADMapping 1105
- GetADStatus 1106
- GetAvailableADGroups 1106
- GetAvailableADUsers 1107
- GetCMSGroupsToSync 1108
- GetCMSRelationshipsToSync 1108
- GetCMSUsersToSync 1109
- GetDomains 1109
- MapCMSUserGroupToAD 1110
- MapCMSUserToAD 1110
- UpdateADMapping 1111
- UpdateADUsersGroups 1111
- UpdateUsersInfoFromAD 1112

ActiveTopics class 15
- about ActiveTopics class 15
- ActiveTopics methods 16
- ActiveTopics properties 17
- BoardID 17
- CacheInterval 17
- DisplayMode 18
- DynamicForumParameter 18
- DynamicGroupParameter 18
- DynamicThreadParameter 19
- Fill 16
- MaxNumber 19
- ToString 16
- URLPath 19
- Wrap 16

Activity class 996
- about Activity class 996
- Activity 996
- Activity methods 997
- Activity properties 1000
- Add 997
- AddActivityMessage 997
- Delete 998
- DisableActivityPublishing 998
- EnableActivityPublishing 998
- GetItem 998
- GetList 999
- GetList method 998
- GetListForUser 999
- IsActivityPublishingEnabled 1001
- Publish 1000
- Update 1000

UpdateActivityMessage 1000
ActivityQueue class 1001
about ActivityQueue class 1001
ActivityStream class 20, 1001
about ActivityStream class 20, 1001
ActivityStream 1001
ActivityStream methods 20, 1002
ActivityStream properties 21
CacheInterval 21
DefaultObjectID 22
DisplayXslt 22
DynamicObjectParameter 22
EnablePaging 23
Fill 20
GetUserActivityStream 1002
GetUserActivityStream method 1002
LoadData 20
MaxResults 23
ObjectType 24
ProfileParamName 24
RaiseCallbackEvent 21
SendActivityToAllStreams 1003
SendActivityToStreams 1003
SendActivityToUsersStream 1003
Stylesheet 24
TemplateUserProfile 25
XmlDoc 25
ActivityType class 1004
about ActivityType class 1004
ActivityType methods 1004
Add 1004
Delete 1004
GetItem 1005
GetList 1005
Update 1005
AddressApi class 602
about AddressApi class 602
Add 603
Add method 603
AddressApi methods 602
Delete 604
GetItem 604
GetList 604
Update 604
UpdateOrderAddress 605
AddressData class 1428
about AddressData class 1428
AddressData properties 1429
AddressLine1 1429
AddressLine2 1429
City 1429
Company 1430
Country 1430
Id 1430
Name 1430
Phone 1430
PostalCode 1430
Region 1431
AdjustedBasketItem class 1431
about AdjustedBasketItem class 1431
AdjustedBasketItem 1431, 1432
AdjustedBasketItem constructor 1431
AdjustedBasketItem properties 1432
AdjustedTotal 1432
ApplicableTaxRate 1432
AppliedCoupons 1433
HasFreeShipping 1433
TaxesApplied 1433
TotalCouponDiscount 1433
Admin enumeration member 1082
ADUser class 951
about ADUser class 951
AddADGroupToCMS 952
AddADUsersToCMSByUsername 952
ADUser methods 951
DeSynchUserGroups 952
DeSynchUsers 953
GetADMapping 953

GetADStatus 953
GetAvailableADGroups 954
GetAvailableADUsers 954
GetCMSGroupsToSync 954
GetCMSRelationshipsToSync 955
GetCMSUsersToSync 955
GetDomains 955
MapCMSUserGroupToAD 956
MapCMSUserToAD 956
SynchCMSGroupsToAD 956
SynchCMSRelationshipsToAD 957
SynchCMSUsersToAD 957
UpdateADMapping 957
UpdateADUsersGroups 958
UpdateUsersInfoFromAD 958
AdvancedEditTemplate class 549
about AdvancedEditTemplate class 549
AdvancedFormBase class 549
about AdvancedFormBase class 549
AdvancedInsertTemplate class 550
about AdvancedInsertTemplate class 550
AdvancedTemplate class 550
about AdvancedTemplate class 550
Analytics class 745, 1133
about Analytics class 745, 1133
Analytics methods 746, 1134
Analytics properties 748
AnalyticsEnabled 748
AppendToIISLog 749
GetAbsoluteUniqueVisitorsTrend 1135, 1136
GetAbsoluteUniqueVisitorsTrend method 1135
GetAdVersions 1137
GetAdVersions method 1136
GetAllTrafficSources 1138, 1139
GetAllTrafficSources method 1138
GetAveragePageViewsTrend 1140
GetAveragePageViewsTrend method 1139
GetBounceRateTrend 1141
GetBounceRateTrend method 1141
GetBrowsers 1142, 1143
GetBrowsers method 1142
GetCampaigns 1144
GetCampaigns method 1143
GetCmsSearchTerms 1145
GetConnectionSpeeds 1146
GetConnectionSpeeds method 1145
GetContentbyTitle 1147, 1148
GetContentbyTitle method 1147
GetContentDetail 1149
GetContentDetail method 1148
GetDirectTraffic 1150, 1151
GetDirectTraffic method 1150
GetFlashVersions 1151, 1152
GetFlashVersions method 1151
GetHostnames 1153
GetHostnames method 1153
GetJavaSupport 1154, 1155
GetJavaSupport method 1154
GetKeywords 1156
GetKeywords method 1155
GetLanguages 1157, 1158
GetLanguages method 1157
GetLocations 1159
GetLocations method 1158
GetNetworkLocations 1160, 1161
GetNetworkLocations method 1160
GetNewVsReturningVisitors 1161, 1162
GetNewVsReturningVisitors method 1161
GetOperatingSystems 1163
GetOperatingSystems method 1163
GetPageViewsTrend 1164, 1165
GetPageViewsTrend method 1164
GetPlatforms 1166
GetPlatforms method 1165
GetReferringSites 1167, 1168
GetReferringSites method 1167
GetScreenColors 1168, 1169
GetScreenColors method 1168

GetScreenResolutions 1170
GetScreenResolutions method 1170
GetSearchEngines 1171, 1172
GetSearchEngines method 1171
GetSessionRank 746
GetTimeOnSiteTrend 1173
GetTimeOnSiteTrend method 1172
GetTopContent 1174, 1175
GetTopContent method 1174
GetTopExitPages 1176
GetTopExitPages method 1175
GetTopLandingPages 1177, 1178
GetTopLandingPages method 1177
GetUserDefinedValue 1178, 1179
GetUserDefinedValue method 1178
GetVisitsTrend 1180
GetVisitsTrend method 1180
HardDumpAnalyticsBuffer 746
InMemoryContentHitsData 749
QueryAnalytics 746
RecordsBeforeWrite 749
SoftDumpAnalyticsBuffer 747
TimeBeforeWrite 749
TrackVisit 747, 748
TrackVisit method 747
AnalyticsTracker class 25
about AnalyticsTracker class 25
AnalyticsTracker methods 26
AnalyticsTracker properties 26
DefaultContentID 26
DynamicParameter 27
EnableAnalytics 27
Wrap 26
AppliedCouponData class 1433
about AppliedCouponData class 1433
AppliedCouponData 1434
AppliedCouponData constructor 1434
AppliedCouponData properties 1434
Coupon 1434
CouponDiscount 1435
AppliedCouponDataCollection class 1435
about AppliedCouponDataCollection class 1435
AppliedCouponDataCollection fields 1435
AppliedCouponDataCollection methods 1435
FindByCouponId 1436
ItemAdded 1435
AppliedCouponEventArgs class 1436
about AppliedCouponEventArgs class 1436
AppliedCoupon 1437
AppliedCouponEventArgs 1436, 1437
AppliedCouponEventArgs constructor 1436
AppliedCouponEventArgs properties 1437
AppointmentTemplate class 550
about AppointmentTemplate class 550
Asset class 822, 1112
about Asset class 822, 1112
Asset methods 823, 1113
GetAssetMgtConfigInfo 823, 1113
GetAssetMgtInfo 823, 1113
GetAssetSupertypes 824, 1114
GetEditSnippet 824, 1114
GetMyAssetMgtID 824, 1115
GetViewUrl 825
MakeCopyDMSAsset 825, 1115
SetAssetMgtConfigInfo 825, 1115
AssetControl class 27
about AssetControl class 27
AssetControl methods 28
AssetControl properties 28
DefaultAssetID 29
DefaultFolderID 29
DynamicParameter 30
Fill 28
IsImage 30
OverrideExtension 30
TaxonomyID 31
ToString 28
UploadType 31

AssetUpdateData class 1606
about AssetUpdateData class 1606
AssetId 1607
AssetUpdateData properties 1607
Comment 1607
ContentId 1608
EndDate 1608
EndDateAction 1608
FileName 1609
FolderId 1609
GoLive 1609
LanguageId 1609
MetaData 1610
TaxonomyTreeIds 1610
Teaser 1610
Title 1610

B

Basket class 1437
about Basket class 1437
AddProduct 1438, 1439
AddProduct method 1438
AddProductVariant 1439
Basket methods 1438
Basket properties 1442
Decrement 1439
GetItemIndex 1440
GetKitConfiguration 1440
GetKitConfigurationAsAdmin 1440
IncrementItemQuantity 1440
MakeDefaultBasket 1440
RemoveItem 1441
RequestInformation 1442
SaveBasket 1441
UpdateItemQuantity 1441
UpdateKitConfiguration 1441
BasketApi class 605
about BasketApi class 605
BasketApi methods 605
CalculateBasket 606, 607
CalculateBasket method 606
CreateNewBasket 607
Delete 608
Empty 608
GetDefaultBasket 609
GetDefaultBasket method 608
GetItem 609
GetList 610
SetDefaultBasket 610, 611
SetDefaultBasket method 610
BasketCalculatorData class 1442
about BasketCalculatorData class 1442
AdjustedItems 1444
AllBasketCoupons 1444
Basket 1444
BasketCalculatorData 1443
BasketCalculatorData constructor 1442
BasketCalculatorData properties 1443
BasketCoupons 1444
BasketTotal 1444
HasFreeShipping 1444
ShipFrom 1445
ShippingRates 1445
ShipTo 1445
TaxClassSubtotals 1445
TotalCouponDiscount 1445
TotalShippingCost 1446
TotalTaxes 1446
ValidationResults 1446
BasketData class 1446
about BasketData class 1446
BasketData properties 1446
DateCreated 1447
DateModified 1447
Id 1447
IsDefault 1447
Items 1447
Name 1448

ShippingAddressId 1448
ShippingMethodId 1448
ShopperId 1448
Subtotal 1448
UserId 1449
BasketItem class 1449
about BasketItem class 1449
AdjustedTotal 1450
BasketItem 1449
BasketItem constructor 1449
BasketItem properties 1450
Configuration 1450
Dimensions 1451
Id 1451
IsInventoryManagementDisabled 1451
IsTangible 1451
ListPrice 1451
Pricing 1452
ProductId 1452
ProductLanguage 1452
ProductSku 1452
ProductTitle 1452
ProductType 1453
Quantity 1453
Quicklink 1453
SalePrice 1453
TaxClassId 1453
Variant 1453
Weight 1454
BasketItemCollection class 1454
about BasketItemCollection class 1454
AddRange 1455
BasketItemCollection 1454
BasketItemCollection methods 1454
Find 1455
FindAll 1455
FindIndex 1455
BasketVariantData class 1455
about BasketVariantData class 1455
BasketVariantData 1456
BasketVariantData properties 1456
Dimensions 1456
Id 1456
Language 1457
ListPrice 1457
Pricing 1457
SalePrice 1457
Sku 1457
TaxClassId 1458
Title 1458
Weight 1458
Blog class 32, 826, 1116
about Blog class 32, 826, 1116
AddBlog 827, 1117
AddCommentForPost 827, 1118
AddPost 827, 1119
ArchiveMode 33
Blog methods 32, 826, 1117
Blog properties 33
BlogID 34
CacheInterval 34
DateToStart 35
DefaultUserID 35
DeleteBlog 829, 1119
DeleteBlogPostComment 830, 1120
DeletePost 830, 1120
DynamicParameter 35
DynamicUserParameter 36
EditorHeight 36
EditorWidth 36
Fill 32
GetBlankBlogPostData 830, 1120
GetBlog 831, 833, 1121, 1122
GetBlog method 830, 1121
GetBlogData 835, 836, 1123, 1124, 1125
GetBlogData method 834, 1123
GetBlogDataForPost 837, 1126
GetBlogPostData 837, 1127

GetBlogRoll 838, 1128
GetBlogRoll method 837, 1127
GetPost 839, 1129
GetPostbyID 839, 1129
GetUserBlog 839, 1130
JavascriptEditorHTMLMode 37
MaxResults 37
PostParameter 37
RecentPosts 38
ShowHeader 38
ShowRSS 38
ToString 32
UpdateBlogProperties 840, 1130
Wrap 33
BlogArchive class 39
about BlogArchive class 39
ArchiveMode 40
BlogArchive methods 39
BlogArchive properties 40
BlogID 40
CacheInterval 41
DynamicParameter 41
Fill 39
Text 41
Wrap 39
WrapTag 42
BlogCalendar class 42
about BlogCalendar class 42
BlogCalendar methods 42
BlogCalendar properties 43
BlogID 43
CacheInterval 43
Fill 42
Text 44
Wrap 42
WrapTag 44
BlogCategories class 44
about BlogCategories class 44
BlogCategories methods 45
BlogCategories properties 45
BlogID 45
CacheInterval 46
Fill 45
Text 46
Wrap 45
WrapTag 46
BlogEntries class 47
about BlogEntries class 47
BlogEntries methods 47
BlogEntries properties 49
BlogID 49
BlogPostParameter 50
CacheInterval 50
DateToStart 50
DefaultUserID 51
DisplayXslt 51
DynamicParameter 51
DynamicUserParameter 52
EditorHeight 52
EditorWidth 52
Fill 47
GetRssFeed 47
GetView 48
GetViewNames 48
JavascriptEditorHTMLMode 53
MaxResults 53
ShowHeader 54
ToString 48
Wrap 48
XmlDoc 54
BlogPost class 54
about BlogPost class 54
BlogPost methods 54
BlogPost properties 55
DefaultContentID 55
ShowType 56
Wrap 55
XmlDoc 56

BlogRecentPosts class 56
about BlogRecentPosts class 56
BlogID 57
BlogRecentPosts methods 57
BlogRecentPosts properties 57
CacheInterval 58
Fill 57
NumberofPosts 58
Text 59
Wrap 57
WrapTag 59
BlogRoll class 59
about BlogRoll class 59
BlogID 60
BlogRoll methods 59
BlogRoll properties 60
CacheInterval 61
Fill 59
Text 61
Wrap 60
WrapTag 61
BlogRSS class 61
about BlogRSS class 61
BlogID 62
BlogRSS methods 62
BlogRSS properties 62
CacheInterval 63
Fill 62
Text 63
WrapTag 63
BreadCrumb class 63
about BreadCrumb class 63
BreadCrumb methods 64
BreadCrumb properties 66
CurrentPageIndicator 66
DefaultContentID 67
DisplayStyle 67
DisplayTitle 67
DynamicParameter 67
Fill 64
GenerateTrail_HTML 64
IconAlt 68
IconPath 68
LinkLastItem 68
LinkTarget 68
MaxItems 69
MetadataName 69
Mode 69
ResetTrail 65
ResetTrail method 64
separator 70
BreadCrumbItem class 70
about BreadCrumbItem class 70
BreadCrumbItem 70, 71
BreadCrumbItem constructor 70
BreadCrumbItem methods 71
BreadCrumbItem properties 72
CaseSensitiveUrlComparison 72
DisplayTitle 72
Equals 71
GetHashCode 71
IconAlt 72
IconPath 73
RawUrl 73
Target 73
ToString 72
BundleData class 1458
about BundleData class 1458
BundleData 1458
BundleData properties 1459
BundledItems 1459
BusinessRule class 671
about BusinessRule class 671
BusinessRule methods 671
GetBusinessRules 672
GetBusinessRules method 671
BusinessRules class 73, 1130
about BusinessRules class 73, 1130

BusinessRules methods 74, 1131
BusinessRules properties 74
Fill 74
GetBusinessRules 1132
GetBusinessRules method 1131
Parameters 75
Result 75
RuleID 75
Text 76
Wrap 74
WrapTag 76

C

Calendar class 76, 1133
about Calendar class 76, 1133
Calendar methods 76
Calendar properties 77
DefaultCalendarID 78
DisplayType 78
DynamicParameter 78
EndDate 79
Fill 77
HighlightEventID 79
HighlightEventTypeID 79
ShowXML 80
StartDate 80
Text 80
Wrap 77
Cart class 81
about Cart class 81
Cart properties 81
DisplayXslt 82
DynamicProductParameter 82
EnableCoupons 82
LoadingImage 83
Settings 83
Stylesheet 83
TemplateCheckout 84
TemplateProduct 84
TemplateShopping 85
UserControlUrl 85
CatalogData class 1459
about CatalogData class 1459
CatalogData properties 1459
Id 1460
Name 1460
CatalogEntryApi class 611
about CatalogEntryApi class 611
Add 612
AddAndCheckIn 612
AddAndPublish 612
AddAndSave 612
CatalogEntryApi methods 611
Checkout 613
Delete 613
GetAttributeEntryList 613
GetItem 614
GetItem method 613
GetItemEdit 614
GetList 615
GetTaxonomyEntryList 615
Restore 615
UndoCheckOut 615
Update 616
UpdateAndCheckIn 616
UpdateAndPublish 616
Checkout class 85
about Checkout class 85
Cart 88
Checkout delegates 94
Checkout enumerations 86
Checkout events 95
Checkout methods 86
Checkout properties 87
CurrentPhase 88
Customer 88
DefaultCountryId 89
DisplayXslt 89

DynamicPhaseParameter 89
Fill 86
FriendlyErrorMessage 90
IsSSLRequired 90
LoadingImage 91
PaymentMethod 91
PhaseChange 95
PhaseChangeEventHandler 94
PostLogin 95
PostLoginEventHandler 95
ShowErrorDetails 91
Stylesheet 92
TemplateCart 92
TemplateOrderHistory 92
TemplateRecoverPassword 93
TemplateShopping 93
ToString 86
UseMsAjax 94
ValidatePhase 94
Wrap 87
CmsApi class 1080
about CmsApi class 1080
ApiMode 1081
CmsApi 1080, 1081
CmsApi constructor 1080
CmsApi properties 1081
ContentLanguage 1081
InPreviewMode 1081
RequestInformation 1082
SitePath 1082
UserId 1082
CmsMessage class 1052
_messageService 1053
about CmsMessage class 1052
Add 1054
CmsMessage 1053
CmsMessage constructor 1053
CmsMessage fields 1053
CmsMessage methods 1054
CmsMessage properties 1056
Delete 1054
GetDefaultItemByType 1054
GetItem 1055
GetItemCollection 1055
GetList 1055
MessageService 1056
Update 1055
CmsMessageType class 1056
_messageTypeService 1057
about CmsMessageType class 1056
Add 1058
CmsMessageType 1057
CmsMessageType constructor 1056
CmsMessageType fields 1057
CmsMessageType methods 1057
CmsMessageType properties 1059
Delete 1058
GetItem 1058
GetList 1058
GetTokenList 1059
MessageTypeService 1059
Update 1059
Collection class 95
about Collection class 95
AddText 100
CacheInterval 100
Collection methods 96
Collection properties 99
CollectionDescription 100
CollectionTitle 101
ContentParameter 101
CopyTo 96
Count 101
Current 101
DefaultCollectionID 102
DynamicParameter 102
EkItems 102
EnablePaging 103

Fill 97
GetEnumerator 97
GetHtml 103
GetRssFeed 97
GetView 97
GetViewNames 98
IncludeIcons 103
IsSynchronized 104
LinkTarget 104
MarkupLanguage 104
MaxResults 104
MoveNext 98
Random 105
Reset 98
SelTaxonomyID 105
SuppressWrapperTags 105
SyncRoot 106
this 98
ToString 98
Wrap 98
XmlDoc 106
CommerceSettings class 1589
about CommerceSettings class 1589
CommerceSettings properties 1590
ComplianceMode 1590
CurrencyAlphaISO 1590
CurrencyCultureCode 1590
CurrencyId 1591
CurrencySymbol 1591
DefaultCurrencyId 1591
ISOCurrencySymbol 1591
OrderProcessingDisabled 1592
PasswordHistory 1592
TestMode 1592
CommunityDocuments class 106
about CommunityDocuments class 106
AddItemFolderId 107
CommunityDocuments methods 106
CommunityDocuments properties 107
DefaultObjectID 108
DynamicContentBoxHeight 108
DynamicContentBoxWidth 108
DynamicContentTemplate 109
EnableAjax 109
EnableSearch 109
Fill 107
MemberMenuActive 109
ObjectType 110
Recursive 110
ShowAllChecked 110
Stylesheet 110
SuppressWrapperTags 111
TaxonomyDepth 111
TaxonomyId 111
CommunityGroup class 750, 1181
about CommunityGroup class 750, 1181
AcceptGroupInvite 752
AddCommunityGroup 752, 1182
AddUserToCommunityGroup 753, 1183, 1184
AddUserToCommunityGroup method 752, 1183
ApprovePendingGroupUser 753, 1184
CancelJoinRequestForCommunityGroup 754, 1185
CommunityGroup methods 751, 1181
DeleteCommunityGroupByID 754, 1185
DeletePendingGroupUser 754, 1186
GetAllCommunityGroups 754, 1186
GetAllCommunityGroupsInDirectory 755, 1186
GetAllUnassignedCommunityGroupUsers 755, 1187
GetCommunityGroupBlog 755
GetCommunityGroupBlogIds 756, 1187
GetCommunityGroupByID 756, 1188
GetCommunityGroupDiscussionBoardIds 756, 1188
GetCommunityGroups 757, 1189
GetCommunityGroupUserIDs 757, 1189
GetCommunityGroupUsers 758, 1190, 1191
GetCommunityGroupUsers method 757, 1190
GetFolderIdForCommunityGroup 758, 1191
GetGroupMemberStatus 759, 1192

GetInvitedCommunityGroups 759, 1192
GetList 759
GetMostPopularCommunityGroups 760, 1193, 1194
GetMostPopularCommunityGroups method 760, 1193
GetPendingCommunityGroups 761, 1194
GetPendingCommunityGroupUsers 761, 1195
InviteUsers 761, 1195
IsCommunityGroupAdmin 762, 1196
IsUserInCommunityGroup 762, 1196
RemoveUserFromCommunityGroup 762, 1197
UpdateCommunityGroup 763, 1197
CommunityGroupBrowser class 111
about CommunityGroupBrowser class 111
AvatarHeight 113
AvatarWidth 113
CacheInterval 114
CommunityGroupBrowser methods 112
CommunityGroupBrowser properties 113
DisplayXslt 114
Fill 112
Link 115
LinkTarget 115
MarkupLanguage 115
MaxResults 115
Stylesheet 116
TagTemplate 116
TagTemplateTarget 116
TaxonomyId 116
ToString 112
Wrap 112
XmlDoc 117
CommunityGroupList class 117
about CommunityGroupList class 117
AllowGroupCreation 119
AvatarHeight 119
AvatarWidth 119
CacheInterval 119
CommunityGroupList methods 117
CommunityGroupList properties 118
DefaultUserID 120
DisplayMode 120
DisplayXslt 120
DynamicUserParameter 121
Fill 118
LanguageId 121
Link 122
LinkTarget 122
MarkupLanguage 122
MaxResults 122
ShowMySortingOptions 123
ShowSortingOptions 123
Stylesheet 123
TagTemplate 123
TagTemplateTarget 124
ToString 118
Wrap 118
XmlDoc 124
CommunityGroupMembers class 124
about CommunityGroupMembers class 124
AvatarHeight 126
AvatarWidth 126
CacheInterval 126
CommunityGroupMembers methods 125
CommunityGroupMembers properties 125
DefaultCommunityGroupID 127
DisplayMode 127
DisplayXslt 127
DynamicParameter 128
Fill 125
Link 128
LinkTarget 128
MarkupLanguage 129
MaxResults 129
Stylesheet 129
ToString 125
Wrap 125
XmlDoc 130
CommunityGroupProfile class 130

about CommunityGroupProfile class 130
CommunityGroupProfile methods 130
CommunityGroupProfile properties 131
DefaultGroupID 131
DisplayXslt 132
DynamicParameter 132
EkItem 132
Fill 130
GroupAdminLink 133
GroupAdminLinkTarget 133
Stylesheet 133
TagTemplate 133
TagTemplateTarget 134
Text 134
ToString 130
Wrap 131
XmlDoc 134
CommunitySearch class 135
about CommunitySearch class 135
CacheInterval 137
CommunitySearch methods 135
CommunitySearch properties 136
CustomPropertyID 138
DisplayXslt 138
EnableFilteringResults 139
EnableGroupResults 139
EnableMap 139
EnableMouseOutHidesInfo 140
EnableMouseOver 140
EnableUnauthorizedSearching 140
EnableUserResults 141
Fill 135
FriendsOnly 141
GroupTaxonomyID 141
InfoBoxShowsEmail 142
LoadingImage 142
MapControl 142
MarkupLanguage 143
MaxTagCount 143
MembersOnly 143
PageSize 144
ResultProfile 144
SearchPrivate 144
ShowNameEmail 145
StartingTab 145
Stylesheet 146
TemplateGroupProfile 146
TemplateParamName 146
TemplateTarget 147
TemplateUserProfile 147
ToString 135
UseMsAjax 147
UserTaxonomyID 148
Wrap 136
XmlDoc 148
XsltDoc 148
Content class 840, 1048, 1198
about Content class 840, 1048, 1198
Add 1048
AddAsset 842
AddContent 843, 845, 846, 1200, 1201, 1202
AddContent method 842, 1199
AddXmlConfiguration 847
CheckOutContent 847, 1203
Content 1048
Content methods 840, 1048, 1198
CopyContentToFolder 847, 1203
Delete 1049
DeleteContentItem 848, 1204
DeleteXmlConfiguration 849
EditAsset 849
GetAddViewLanguage 849, 1204
GetAllComments 850, 1205
GetAllTemplates 850, 1205
GetAllUnassignedItemApprovals 850, 1206
GetAllXmlConfigurations 851
GetAllXmlConfigurations method 851
GetChildContent 852, 853, 1206, 1207

GetChildContent method 852, 1206
GetCommunityTemplate 854
GetContent 855, 856, 1208
GetContent method 854
GetContentByHistory 857, 1208
GetContentDisplayLink 858
GetContentForEditing 858, 1209
GetContentReport 859
GetContentReport method 858
GetContentState 859, 1209
GetContentStatus 860, 1210
GetDomain 860, 1210
GetExpireContent 860
GetItem 1049
GetList 861, 1049
GetList method 860
MoveContentToFolder 861, 1210
PublishContent 862, 1211
RelicateXmlConfiguration 862
RenameContent 863, 1211
ReplicateXmlConfiguration 863
RestoreHistoryContent 863
SaveContent 863, 1212
SubmitForDelete 864, 1213
UndoCheckout 864, 1213
Update 1049
UpdateContentMetaData 865, 1213
UpdatexmlCollectionPackage 865
UpdateXmlConfiguration 865
ContentBlock class 148
about ContentBlock class 148
CacheInterval 151
ContentBlock methods 149
ContentBlock properties 151
CopyTo 149
Count 152
Current 152
DefaultContentID 152
DisplayXslt 153
DynamicParameter 153
EkItem 153
Fill 150
GetEnumerator 150
IsSynchronized 154
MoveNext 150
OverrideXslt 154
Reset 150
StartingFolderID 154
SyncRoot 155
this 150
ToString 150
Wrap 151
ContentFlagging class 155
about ContentFlagging class 155
ContentFlagging methods 155
ContentFlagging properties 156
DefaultContentID 156
DisplayXslt 157
DynamicParameter 157
Fill 155
FlagImage 157
FlagText 158
Stylesheet 158
ToString 155
Wrap 156
XmlDoc 159
ContentInfo class 558
about ContentInfo class 558
AssetID 558
Comments 559
ContentInfo properties 558
ErrorMessage 559
FileAndPath 559
FileExtension 559
FileName 559
FolderID 559
Html 560
ID 560

Language 560
MimeType 560
Status 560
Title 560
Type 561
UserID 561
ContentList class 159
about ContentList class 159
CacheInterval 163
ContentIds 163
ContentList methods 159
ContentList properties 162
CopyTo 160
Count 164
Current 164
DefaultContentID 164
Direction 164
DynamicParameter 165
EkItems 165
Fill 160
GetEnumerator 160
GetHtml 165
GetRssFeed 160
IncludeIcons 166
IsSynchronized 166
LinkTarget 166
MarkupLanguage 167
MetaTag 167
MoveNext 160
OrderByDirection 167
OrderKey 168
Random 168
Reset 161
SuppressWrapperTags 168
SyncRoot 169
this 161
ToString 161
Wrap 161
XmlDoc 169
ContentRating class 169, 866, 1214
about ContentRating class 169, 866, 1214
AddContentRating 1215
AlreadyRatedMessage 171
BadLabel 171
CharsRemainingLabel 172
CompleteGraphDisplay 172
ContentRating methods 169, 866, 1214
ContentRating properties 170
CookiesRequiredMessage 172
DefaultContentID 173
DynamicParameter 173
FeedbackHeader 173
Fill 170
GetContentRating 866
GetContentRatingAddRating 868
GetContentRatingResults 869, 1215
GetContentRatingStatistics 869, 870
GetContentRatingStatistics method 869
GoodLabel 174
GraphBarColor 174
InitialGraphDisplay 174
Padding 175
PurgeContentRatings 870, 1216
RatingLevelLabel 175
ThankYouMessage 176
ToString 170
TotalRatingsLabel 176
UserCommentsHeader 176
ValidationMessage 177
Wrap 170
ContentRatingLabels class 1611
about ContentRatingLabels class 1611
AlreadyRatedMessage 1611
BadLabel 1612
CharsRemainingLabel 1612
ContentRatingLabels properties 1611
CookiesRequiredMessage 1612
FeedbackHeader 1612

GoodLabel 1613
RatingLevelLabel 1613
ThankYouMessage 1613
TotalRatingsLabel 1613
UserCommentsHeader 1613
ValidationMessage 1614
ContentReview class 177
about ContentReview class 177
ContentReview methods 177
ContentReview properties 179
DefaultContentID 179
DisplayXslt 180
DynamicParameter 180
DynamicUserParameter 181
Fill 178
GetReviews 181
MaxReviews 181
Moderate 182
RaiseCallbackEvent 178
RatingsMinimum 182
ToString 178
UserID 182
Wrap 178
XmlDoc 183
ContractApi class 616
about ContractApi class 616
Add 617
ContractApi methods 617
Delete 617
GetItem 617
GetList 618
Update 618
ContractData class 1460
about ContractData class 1460
ContractData 1461
ContractData constructor 1460
ContractData properties 1461
Id 1461
Name 1461
CountryApi class 618
about CountryApi class 618
Add 619
CanDelete 619
CountryApi methods 618
Delete 619
GetItem 620
GetList 620
Update 620
CountryData class 1461
about CountryData class 1461
CountryData 1462
CountryData constructor 1462
CountryData methods 1463
CountryData properties 1463
Enabled 1463
Id 1464
LongIsoCode 1464
Name 1464
ShortIsoCode 1464
ToString 1463
Validate 1463
CountryTaxRateData class 1464
about CountryTaxRateData class 1464
CountryData 1466
CountryTaxRateData 1465
CountryTaxRateData constructor 1465
CountryTaxRateData properties 1466
CouponApi class 620
about CouponApi class 620
Add 621
AddCouponToCatalog 622
AddCouponToProduct 622
AddCouponToTaxonomy 622
ApplyCouponToBasket 623
CalculateBasketCoupons 623
CalculateBasketCoupons method 623
CouponApi methods 621
Deactivate 624

Delete 624
DeleteCouponProducts 624
GetCatalogList 624
GetCouponsAppliedToBasket 625
GetItem 625
GetList 625
GetProductList 626
GetProductList method 625
GetTaxonomyList 626
IsCouponAppliedToProduct 626
IsCouponUsedForBasket 627
IsCouponUsedForOrder 627
SaveCouponApplications 627
Update 628
ValidateCoupon 628
ValidateCoupon method 628
CouponData class 1466
about CouponData class 1466
Code 1467
CouponData 1466
CouponData properties 1466
CouponType 1467
CurrencyId 1467
Description 1468
DiscountType 1468
DiscountValue 1468
ExpirationDate 1468
Id 1468
IsActive 1468
IsCombinable 1469
IsRedeemable 1469
MaximumAmount 1469
MaximumUses 1469
MinimumAmount 1469
OnePerCustomer 1469
StartDate 1470
CouponEntryData class 1470
about CouponEntryData class 1470
CouponEntryData 1470, 1471
CouponEntryData constructor 1470
CouponEntryData properties 1471
CouponId 1471
DataState 1471
IsIncluded 1472
ObjectId 1472
ObjectType 1472
Path 1472
Title 1472
CurrencyApi class 629
about CurrencyApi class 629
Add 629
CanDelete 629
CurrencyApi methods 629
Delete 630
GetItem 630
GetList 630
Update 630
CurrencyData class 1472
about CurrencyData class 1472
AlphaIsoCode 1473
CultureCode 1473
CurrencyData properties 1473
CurrencySymbol 1473
Enabled 1474
Id 1474
ISOCurrencySymbol 1474
Name 1474
CurrencyPricingData class 1474
about CurrencyPricingData class 1474
AlphaIsoCode 1477
Clone 1476
CurrencyId 1477
CurrencyPricingData 1475
CurrencyPricingData constructor 1475
CurrencyPricingData methods 1476
CurrencyPricingData properties 1477
GetSalePrice 1476
ListPrice 1477

NumericIsoCode 1477
PricingType 1478
SalePrice 1478
TierPrices 1478
Validate 1476
CurrencySelect class 183
about CurrencySelect class 183
CacheInterval 184
CurrencySelect methods 183
CurrencySelect properties 184
DisplayXslt 185
EkItem 186
Fill 184
Stylesheet 186
ToString 184
Wrap 184
CustomerApi class 631
about CustomerApi class 631
Add 632
Add method 631
ChangeBillingAddress 632
ChangeShippingAddress 633
CustomerApi methods 631
Delete 633
GetItem 633
GetList 633
GetOrderList 634
GetOrderTotal 634
Update 634
CustomerData class 1478
about CustomerData class 1478
AverageOrderValue 1479
BillingAddressId 1479
CustomerData properties 1478
DisplayName 1479
EmailAddress 1479
FirstName 1480
Id 1480
IsDeleted 1480
IsMembershipUser 1480
LastName 1480
Password 1480
ShippingAddressId 1481
TotalOrders 1481
TotalOrderValue 1481
UserId 1481
UserName 1481
CustomFields class 673, 1216
about CustomFields class 673, 1216
AddNewEditableCustomFieldAssignments 674, 1218
BoolToYesNo 675, 1218
CustomFields methods 673, 1217
GetEditableCustomFieldAssignments 675, 1219
GetEnhancedMetadataArea 676, 1220
GetEnhancedMetadataScript 676, 1220
GetFieldsByFolder 676, 1220
GetSearchProperties 677, 678, 1221
GetSearchProperties method 677
Populate_AssetRequestObjectFromForm 678
ProcessCustomFields 679, 1221
SearchAllAssets 679, 1222
WriteFilteredMetadataForEdit 679, 1222
WriteFilteredMetadataForView 680, 1223
WriteLibrarySearchExtended 680, 1223
WriteMetadataDefaultForEdit 681, 1224
WriteMetadataTypeForView 681, 1224
WritePublicSearch 681, 1225
WritePublicSearchEx 682

D

Data Classes 1423
DataKeys class 551
about DataKeys class 551
DesignTimeDiagnostic class 186
about DesignTimeDiagnostic class 186
Authenticated 187
DesignTimeDiagnostic methods 187
DesignTimeDiagnostic properties 187

Fill 187
Hide 187
Language 188
SuppressWrapperTags 188
WrapTag 188
DhtmlMenu class 188
about DhtmlMenu class 188
AccessKey 190
Align 190
BackColor 191
BorderColor 191
BorderStyle 191
BorderWidth 191
CssClass 192
DhtmlMenu methods 189
DhtmlMenu properties 189
DisplayXslt 192
Enabled 192
EnableIE6FormsFix 192
Fill 189
Height 193
HorizontalDisplay 193
MenuHoverOverColor 193
MenuStyle 194
Stylesheet 194
SubMenuBackColor 194
SubMenuHoverOverColor 194
SubMenuItemHeight 195
SubMenuItemWidth 195
SubMenuTopOffset 195
TabIndex 196
ToolTip 196
Width 196
Wrap 189
Dimensions class 1482
about Dimensions class 1482
Clone 1483
CompareTo 1483
ConvertUnits 1483
CubicUnits 1484
Dimensions 1482
Dimensions constructor 1482
Dimensions methods 1483
Dimensions properties 1484
Height 1484
Length 1484
Units 1484
Width 1484
Directory class 196
about Directory class 196
AddItemFolderId 201
CacheInterval 201
CopyTo 197
Count 202
Current 202
Directory methods 197
Directory properties 200
DisplayXslt 202
DynamicParameter 202
EkItems 203
EnableAjax 203
EnablePaging 203
EnableSearch 204
Fill 197
GetEnumerator 198
GetView 198
GetViewNames 198
IncludeIcons 204
IsSynchronized 204
ItemSortOrder 204
LinkTarget 205
MarkupLanguage 205
MaxResults 205
MoveNext 198
Recursive 206
Reset 199
ShowAllChecked 206
ShowCount 206

ShowRoot 207
SortDirection 207
SuppressWrapperTags 207
SyncRoot 208
TaxonomyCols 208
TaxonomyDepth 208
TaxonomyDescription 208
TaxonomyId 209
TaxonomyItemCols 209
TaxonomyParamId 209
TaxonomyPath 209
TaxonomyTitle 210
TaxonomyTreeData 210
TaxonomyXml 210
this 199
ToString 199
Wrap 199
XmlDoc 210
DiscussionCategory class 1614
about DiscussionCategory class 1614
BoardID 1614
CategoryID 1615
DiscussionCategory properties 1614
Name 1615
SortOrder 1615
DisplayTemplate class 551
about DisplayTemplate class 551

E

eCommerce 601
EkEditorPart class 12
about EkEditorPart class 12
Ektron namespace
Classes 601
Ektron.Cms 7
Ektron.Cms namespace 7
Ektron.Cms.API 669
Ektron.Cms.API namespace 669
Classes 670
Ektron.Cms.API.Analytics 745
Ektron.Cms.API.Analytics namespace 745
Classes 745
Ektron.Cms.API.Analytics.Analytics 745
Ektron.Cms.API.Analytics.Analytics.AnalyticsEnabled 748
Ektron.Cms.API.Analytics.Analytics.AppendToIISLog 749
Ektron.Cms.API.Analytics.Analytics.GetSessionRank 746
Ektron.Cms.API.Analytics.Analytics.HardDumpAnalyticsBuffer 746
Ektron.Cms.API.Analytics.Analytics.InMemoryContentHitsData 749
Ektron.Cms.API.Analytics.Analytics.QueryAnalytics 746
Ektron.Cms.API.Analytics.Analytics.RecordsBeforeWrite 749
Ektron.Cms.API.Analytics.Analytics.SoftDumpAnalyticsBuffer 747
Ektron.Cms.API.Analytics.Analytics.TimeBeforeWrite 749
Ektron.Cms.API.Analytics.Analytics.TrackVisit 747, 748
Ektron.Cms.API.BusinessRule 671
Ektron.Cms.API.BusinessRule.GetBusinessRules 672
Ektron.Cms.API.Community 750
Ektron.Cms.API.Community namespace 750
Classes 750
Ektron.Cms.API.Community.CommunityGroup 750
Ektron.Cms.API.Community.CommunityGroup.AcceptGroupInvite 752
Ektron.Cms.API.Community.CommunityGroup.AddCommunityGroup 752
Ektron.Cms.API.Community.CommunityGroup.AddUserToCommunityGroup 753
Ektron.Cms.API.Community.CommunityGroup.ApprovePendingGroupUser 753
Ektron.Cms.API.Community.CommunityGroup.CancelJoinRequestForCommunityGroup 754
Ektron.Cms.API.Community.CommunityGroup.DeleteCommunityGroupByID 754
Ektron.Cms.API.Community.CommunityGroup.DeletePendingGroupUser 754
Ektron.Cms.API.Community.CommunityGroup.GetAllCommunityGroups

754

Ektron.Cms.API.Community.CommunityGroup.GetAllCommunityGroupsInDirectory
755

Ektron.Cms.API.Community.CommunityGroup.GetAllUnassignedCommunityGroupUsers
755

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroupBlog
755

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroupBlogIds
756

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroupByID
756

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroupDiscussionBoardIds
756

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroups
757

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroupUserIDs
757

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroupUsers
758

Ektron.Cms.API.Community.CommunityGroup.GetFolderIdForCommunityGroup
758

Ektron.Cms.API.Community.CommunityGroup.GetGroupMemberStatus
759

Ektron.Cms.API.Community.CommunityGroup.GetInvitedCommunityGroups
759

Ektron.Cms.API.Community.CommunityGroup.GetList 759

Ektron.Cms.API.Community.CommunityGroup.GetMostPopularCommunityGroups
760

Ektron.Cms.API.Community.CommunityGroup.GetPendingCommunityGroups
761

Ektron.Cms.API.Community.CommunityGroup.GetPendingCommunityGroupUsers
761

Ektron.Cms.API.Community.CommunityGroup.InviteUsers 761

Ektron.Cms.API.Community.CommunityGroup.IsCommunityGroupAdmin
762

Ektron.Cms.API.Community.CommunityGroup.IsUserInCommunityGroup
762

Ektron.Cms.API.Community.CommunityGroup.RemoveUserFromCommunityGroup
762

Ektron.Cms.API.Community.CommunityGroup.UpdateCommunityGroup
763

Ektron.Cms.API.Community.Favorites 763

Ektron.Cms.API.Community.Favorites.AddContentFavorite 764

Ektron.Cms.API.Community.Favorites.AddFavoriteLink 765

Ektron.Cms.API.Community.Favorites.DeleteContentFavorite 765

Ektron.Cms.API.Community.Favorites.DeleteFavoriteLink 765

Ektron.Cms.API.Community.Favorites.GetAllFavoriteLink 766, 767

Ektron.Cms.API.Community.Favorites.GetContentFavorites 768, 769

Ektron.Cms.API.Community.Favorites.GetFavoriteId 769

Ektron.Cms.API.Community.Favorites.GetFavoriteLink 769

Ektron.Cms.API.Community.Favorites.GetFavoritesDirectory 770

Ektron.Cms.API.Community.Favorites.IsContentFavorite 770

Ektron.Cms.API.Community.Favorites.MoveContentFavorite 770

Ektron.Cms.API.Community.Favorites.MoveFavoriteLink 771

Ektron.Cms.API.Community.Favorites.UpdateFavoriteLink 771

Ektron.Cms.API.Community.Flagging 771

Ektron.Cms.API.Community.Flagging.AddFlagObject 772

Ektron.Cms.API.Community.Flagging.AssignFlagToContent 772

Ektron.Cms.API.Community.Flagging.GetAllFlagEntries 773, 774

Ektron.Cms.API.Community.Flagging.GetContentFlagData 774, 775

Ektron.Cms.API.Community.Flagging.GetDefaultFolderFlagDef 775

Ektron.Cms.API.Community.Flagging.GetFlagDataById 775

Ektron.Cms.API.Community.Flagging.GetFlaggedRank 776

Ektron.Cms.API.Community.Flagging.PurgeFlagEntries 776

Ektron.Cms.API.Community.Flagging.RemoveObjectFlag 776

Ektron.Cms.API.Community.Flagging.UpdateFlagObject 777

Ektron.Cms.API.Community.Friends 777

Ektron.Cms.API.Community.Friends.AcceptFriendInvite 778

Ektron.Cms.API.Community.Friends.AcceptPendingFriend 778

Ektron.Cms.API.Community.Friends.AddFriend 779

Ektron.Cms.API.Community.Friends.AddPendingFriend 779

Ektron.Cms.API.Community.Friends.AddSelectedFriend 780

Ektron.Cms.API.Community.Friends.DeleteFriend 780

Ektron.Cms.API.Community.Friends.DeletePendingFriendRequest
780, 781

Ektron.Cms.API.Community.Friends.DeleteSelectedFriend 781

Ektron.Cms.API.Community.Friends.GetFriends 782, 783

Ektron.Cms.API.Community.Friends.GetFriendsDirectory 783

Ektron.Cms.API.Community.Friends.GetFriendStatus 784

Ektron.Cms.API.Community.Friends.GetInvitedUsers 784

Ektron.Cms.API.Community.Friends.GetPendingFriends 784

Ektron.Cms.API.Community.Friends.GetSelectedFriends 785

Ektron.Cms.API.Community.Friends.GetSentFriendInvitations
785

Ektron.Cms.API.Community.Friends.InviteUsers 785

Ektron.Cms.API.Community.Friends.IsFriend 786

Ektron.Cms.API.Community.Friends.IsSelectedFriend 786

Ektron.Cms.API.Community.Friends.MoveFriend 786

Ektron.Cms.API.Community.MessageBoard 787

Ektron.Cms.API.Community.MessageBoard.AddMessageBoardEntry
788

Ektron.Cms.API.Community.MessageBoard.ApproveMessage
789

Ektron.Cms.API.Community.MessageBoard.DeleteMessageBoardEntries
790, 791

Ektron.Cms.API.Community.MessageBoard.DeleteMessageBoardEntriesByUser
792, 793

Ektron.Cms.API.Community.MessageBoard.DeleteMessageBoardEntry
794, 795

Ektron.Cms.API.Community.MessageBoard.DisableModeration
795

Ektron.Cms.API.Community.MessageBoard.EnableModeration
795

Ektron.Cms.API.Community.MessageBoard.GetAllMessageBoardEntries
796, 797

Ektron.Cms.API.Community.MessageBoard.GetAllMessageBoardEntriesByUser
798, 799

Ektron.Cms.API.Community.MessageBoard.GetMessageBoardEntry
800, 801

Ektron.Cms.API.Community.MessageBoard.IsModerated 802

Ektron.Cms.API.Community.MessageBoard.IsSpam 802

Ektron.Cms.API.Community.MessageBoard.IsUserMessageBoardAdmin
802

Ektron.Cms.API.Community.MessageBoard.UpdateMessageText
803, 804

Ektron.Cms.API.Community.Tags 805

Ektron.Cms.API.Community.Tags.AddTag 806

Ektron.Cms.API.Community.Tags.AddTagToCommunityGroup
807

Ektron.Cms.API.Community.Tags.AddTagToContent 808, 809

Ektron.Cms.API.Community.Tags.AddTagToObject 810

Ektron.Cms.API.Community.Tags.AddTagToUser 811, 812

Ektron.Cms.API.Community.Tags.DeleteTagById 812

Ektron.Cms.API.Community.Tags.DeleteTagOnObject 813

Ektron.Cms.API.Community.Tags.DeleteTagsByUser 813

Ektron.Cms.API.Community.Tags.DeleteTagsOnCommunityGroup
813

Ektron.Cms.API.Community.Tags.DeleteTagsOnContent 814

Ektron.Cms.API.Community.Tags.DeleteTagsOnObject 814

Ektron.Cms.API.Community.Tags.DeleteTagsOnUser 814

Ektron.Cms.API.Community.Tags.EditTag 814

Ektron.Cms.API.Community.Tags.GetAllTagRelatedContent 815

Ektron.Cms.API.Community.Tags.GetAllTagRelatedUsers 815

Ektron.Cms.API.Community.Tags.GetAllTags 816

Ektron.Cms.API.Community.Tags.GetDefaultTags 817

Ektron.Cms.API.Community.Tags.GetTagByID 818

Ektron.Cms.API.Community.Tags.GetTagByName 818

Ektron.Cms.API.Community.Tags.GetTagCloud 818

Ektron.Cms.API.Community.Tags.GetTagsForContent 818

Ektron.Cms.API.Community.Tags.GetTagsForObject 819, 820

Ektron.Cms.API.Community.Tags.GetTagsForUser 820, 821

Ektron.Cms.API.Community.Tags.RemoveTagAsDefault 821

Ektron.Cms.API.Community.Tags.SaveTagAsDefault 821

Ektron.Cms.API.Content 822

Ektron.Cms.API.Content namespace 822

Classes 822

Ektron.Cms.API.Content.Asset 822
Ektron.Cms.API.Content.Asset.GetAssetMgtConfigInfo 823
Ektron.Cms.API.Content.Asset.GetAssetMgtInfo 823
Ektron.Cms.API.Content.Asset.GetAssetSupertypes 824
Ektron.Cms.API.Content.Asset.GetEditSnippet 824
Ektron.Cms.API.Content.Asset.GetMyAssetMgtID 824
Ektron.Cms.API.Content.Asset.GetViewUrl 825
Ektron.Cms.API.Content.Asset.MakeCopyDMSAsset 825
Ektron.Cms.API.Content.Asset.SetAssetMgtConfigInfo 825
Ektron.Cms.API.Content.Blog 826
Ektron.Cms.API.Content.Blog.AddBlog 827
Ektron.Cms.API.Content.Blog.AddCommentForPost 827
Ektron.Cms.API.Content.Blog.AddPost 827
Ektron.Cms.API.Content.Blog.DeleteBlog 829
Ektron.Cms.API.Content.Blog.DeleteBlogPostComment 830
Ektron.Cms.API.Content.Blog.DeletePost 830
Ektron.Cms.API.Content.Blog.GetBlankBlogPostData 830
Ektron.Cms.API.Content.Blog.GetBlog 831, 833
Ektron.Cms.API.Content.Blog.GetBlogData 835, 836
Ektron.Cms.API.Content.Blog.GetBlogDataForPost 837
Ektron.Cms.API.Content.Blog.GetBlogPostData 837
Ektron.Cms.API.Content.Blog.GetBlogRoll 838
Ektron.Cms.API.Content.Blog.GetPost 839
Ektron.Cms.API.Content.Blog.GetPostbyID 839
Ektron.Cms.API.Content.Blog.GetUserBlog 839
Ektron.Cms.API.Content.Blog.UpdateBlogProperties 840
Ektron.Cms.API.Content.Content 840
Ektron.Cms.API.Content.Content.AddAsset 842
Ektron.Cms.API.Content.Content.AddContent 843, 845, 846
Ektron.Cms.API.Content.Content.AddXmlConfiguration 847
Ektron.Cms.API.Content.Content.CheckOutContent 847
Ektron.Cms.API.Content.Content.CopyContentToFolder 847
Ektron.Cms.API.Content.Content.DeleteContentItem 848
Ektron.Cms.API.Content.Content.DeleteXmlConfiguration 849
Ektron.Cms.API.Content.Content.EditAsset 849
Ektron.Cms.API.Content.Content.GetAddViewLanguage 849
Ektron.Cms.API.Content.Content.GetAllComments 850
Ektron.Cms.API.Content.Content.GetAllTemplates 850
Ektron.Cms.API.Content.Content.GetAllUnassignedItemApprovals 850
Ektron.Cms.API.Content.Content.GetAllXmlConfigurations 851
Ektron.Cms.API.Content.Content.GetChildContent 852, 853
Ektron.Cms.API.Content.Content.GetCommunityTemplate 854
Ektron.Cms.API.Content.Content.GetContent 855, 856
Ektron.Cms.API.Content.Content.GetContentByHistory 857
Ektron.Cms.API.Content.Content.GetContentDisplayLink 858
Ektron.Cms.API.Content.Content.GetContentForEditing 858
Ektron.Cms.API.Content.Content.GetContentReport 859
Ektron.Cms.API.Content.Content.GetContentState 859
Ektron.Cms.API.Content.Content.GetContentStatus 860
Ektron.Cms.API.Content.Content.GetDomain 860
Ektron.Cms.API.Content.Content.GetExpireContent 860
Ektron.Cms.API.Content.Content.GetList 861
Ektron.Cms.API.Content.Content.MoveContentToFolder 861
Ektron.Cms.API.Content.Content.PublishContent 862
Ektron.Cms.API.Content.Content.RelicateXmlConfiguration 862
Ektron.Cms.API.Content.Content.RenameContent 863
Ektron.Cms.API.Content.Content.ReplicateXmlConfiguration 863
Ektron.Cms.API.Content.Content.RestoreHistoryContent 863
Ektron.Cms.API.Content.Content.SaveContent 863
Ektron.Cms.API.Content.Content.SubmitForDelete 864
Ektron.Cms.API.Content.Content.UndoCheckout 864
Ektron.Cms.API.Content.Content.UpdateContentMetaData 865
Ektron.Cms.API.Content.Content.UpdatexmlCollectionPackage 865
Ektron.Cms.API.Content.Content.UpdateXmlConfiguration 865
Ektron.Cms.API.Content.ContentRating 866
Ektron.Cms.API.Content.ContentRating.GetContentRating 866
Ektron.Cms.API.Content.ContentRating.GetContentRatingAddRating 868
Ektron.Cms.API.Content.ContentRating.GetContentRatingResults 869
Ektron.Cms.API.Content.ContentRating.GetContentRatingStatistics 869, 870
Ektron.Cms.API.Content.ContentRating.PurgeContentRatings 870
Ektron.Cms.API.Content.Form 871

Ektron.Cms.API.Content.Form.AssignFormTask 871
Ektron.Cms.API.Content.Form.EmailFormFieldData 872
Ektron.Cms.API.Content.Form.GetAllForms 873
Ektron.Cms.API.Content.Form.GetForm 873
Ektron.Cms.API.Content.Form.GetFormFieldData 874
Ektron.Cms.API.Content.Form.GetFormFieldList 874
Ektron.Cms.API.Content.Form.GetFormFieldQuestions 875
Ektron.Cms.API.Content.Form.GetFormTitle 875
Ektron.Cms.API.Content.Form.SubmitFormFieldData 875
Ektron.Cms.API.Content.Form.UpdateFormFieldData 876
Ektron.Cms.API.Content.Taxonomy 876
Ektron.Cms.API.Content.Taxonomy.AddTaxonomyItem 877, 878
Ektron.Cms.API.Content.Taxonomy.AddTaxonomySyncFolder 878
Ektron.Cms.API.Content.Taxonomy.CloneTaxonomy 878
Ektron.Cms.API.Content.Taxonomy.CreateTaxonomy 879
Ektron.Cms.API.Content.Taxonomy.DeleteTaxonomy 879
Ektron.Cms.API.Content.Taxonomy.GetTaxonomyIdByPath 879
Ektron.Cms.API.Content.Taxonomy.ImportTaxonomy 880
Ektron.Cms.API.Content.Taxonomy.LoadTaxonomy 880
Ektron.Cms.API.Content.Taxonomy.ReadAllAssignedCategory 881
Ektron.Cms.API.Content.Taxonomy.ReadTaxonomy 881
Ektron.Cms.API.Content.Taxonomy.RemoveTaxonomyFolder 881
Ektron.Cms.API.Content.Taxonomy.RemoveTaxonomyItem 882
Ektron.Cms.API.Content.Taxonomy.ReOrderTaxonomyItems 882
Ektron.Cms.API.Content.Taxonomy.UpdateTaxonomy 882
Ektron.Cms.API.Content.ThreadedDiscussion 882
Ektron.Cms.API.Content.ThreadedDiscussion.AddBoard 884
Ektron.Cms.API.Content.ThreadedDiscussion.AddForum 884
Ektron.Cms.API.Content.ThreadedDiscussion.AddReply 887
Ektron.Cms.API.Content.ThreadedDiscussion.AddTopic 888, 889
Ektron.Cms.API.Content.ThreadedDiscussion.DeleteBoard 889
Ektron.Cms.API.Content.ThreadedDiscussion.DeleteForum 889
Ektron.Cms.API.Content.ThreadedDiscussion.DeleteTopic 889
Ektron.Cms.API.Content.ThreadedDiscussion.GetActiveTopics 890
Ektron.Cms.API.Content.ThreadedDiscussion.GetActiveTopicsForForum 891
Ektron.Cms.API.Content.ThreadedDiscussion.GetActiveUsers 891
Ektron.Cms.API.Content.ThreadedDiscussion.GetBoardCategories 891
Ektron.Cms.API.Content.ThreadedDiscussion.GetBoardForums 892
Ektron.Cms.API.Content.ThreadedDiscussion.GetDiscussionBoard 892, 893
Ektron.Cms.API.Content.ThreadedDiscussion.GetDiscussionBoardString 893
Ektron.Cms.API.Content.ThreadedDiscussion.GetForum 893
Ektron.Cms.API.Content.ThreadedDiscussion.GetRepliesForTopics 895
Ektron.Cms.API.Content.ThreadedDiscussion.GetTopic 895
Ektron.Cms.API.Content.ThreadedDiscussion.SearchReplies 896
Ektron.Cms.API.Content.ThreadedDiscussion.SubscribeToBoard 896
Ektron.Cms.API.Content.ThreadedDiscussion.SubscribeToForum 897
Ektron.Cms.API.Content.ThreadedDiscussion.SubscribeToTopic 897
Ektron.Cms.API.Content.ThreadedDiscussion.UnSubscribeToBoard 897
Ektron.Cms.API.Content.ThreadedDiscussion.UnSubscribeToForum 898
Ektron.Cms.API.Content.ThreadedDiscussion.UnSubscribeToTopic 898
Ektron.Cms.API.CustomFields 673
Ektron.Cms.API.CustomFields.AddNewEditableCustomFieldAssignments 674
Ektron.Cms.API.CustomFields.BoolToYesNo 675
Ektron.Cms.API.CustomFields.GetEditableCustomFieldAssignments 675
Ektron.Cms.API.CustomFields.GetEnhancedMetadataArea 676

Ektron.Cms.API.CustomFields.GetEnhancedMetadataScript 676
Ektron.Cms.API.CustomFields.GetFieldsByFolder 676
Ektron.Cms.API.CustomFields.GetSearchProperties 677, 678
Ektron.Cms.API.CustomFields.Populate_AssetRequestObjectFromForm 678
Ektron.Cms.API.CustomFields.ProcessCustomFields 679
Ektron.Cms.API.CustomFields.SearchAllAssets 679
Ektron.Cms.API.CustomFields.WriteFilteredMetadataForEdit 679
Ektron.Cms.API.CustomFields.WriteFilteredMetadataForView 680
Ektron.Cms.API.CustomFields.WriteLibrarySearchExtended 680
Ektron.Cms.API.CustomFields.WriteMetadataDefaultForEdit 681
Ektron.Cms.API.CustomFields.WriteMetadataTypeForView 681
Ektron.Cms.API.CustomFields.WritePublicSearch 681
Ektron.Cms.API.CustomFields.WritePublicSearchEx 682
Ektron.Cms.API.Folder 683
Ektron.Cms.API.Folder.AddCommunityFolder 684
Ektron.Cms.API.Folder.AddFolder 684
Ektron.Cms.API.Folder.CreateFolder 685
Ektron.Cms.API.Folder.DeleteFolderById 685
Ektron.Cms.API.Folder.GetChildFolders 687, 688
Ektron.Cms.API.Folder.GetFolder 690, 691
Ektron.Cms.API.Folder.GetFolderDataWithPermission 691
Ektron.Cms.API.Folder.GetFolderId 692
Ektron.Cms.API.Folder.GetFolderInfo 693
Ektron.Cms.API.Folder.GetPath 693
Ektron.Cms.API.Folder.IsFolderHidden 693
Ektron.Cms.API.Folder.RenameFolder 694
Ektron.Cms.API.Folder.UpdateFolder 694, 695
Ektron.Cms.API.Folder.UpdateFolderApprovalMethod 695
Ektron.Cms.API.Library 695
Ektron.Cms.API.Library.AddLibraryItem 696
Ektron.Cms.API.Library.AddLoadBalanceItem 696
Ektron.Cms.API.Library.DeleteLibraryItem 697
Ektron.Cms.API.Library.DeleteLibraryItemById 697
Ektron.Cms.API.Library.DeleteLoadBalanceItem 697
Ektron.Cms.API.Library.GetAllChildLibItems 697
Ektron.Cms.API.Library.GetAllLoadBalancePaths 700
Ektron.Cms.API.Library.GetLibraryItem 701
Ektron.Cms.API.Library.GetLibrarySettings 701
Ektron.Cms.API.Library.GetLibraryTypes 702
Ektron.Cms.API.Library.GetLoadBalancePath 702
Ektron.Cms.API.Library.InternalLibrarySearch 702
Ektron.Cms.API.Library.UpdateLibrary 703
Ektron.Cms.API.Library.UpdateLibrarySettings 703
Ektron.Cms.API.Library.UpdateLoadBalanceSettings 703
Ektron.Cms.API.Library.UpdateQlinkTemplates 703
Ektron.Cms.API.Localization 704
Ektron.Cms.API.Localization.GetAddViewLanguage 705
Ektron.Cms.API.Localization.GetAddViewLanguageForAllContent 705
Ektron.Cms.API.Localization.GetAddViewLanguageForMenus 705
Ektron.Cms.API.Localization.GetContentItems 706
Ektron.Cms.API.Localization.GetFile 706
Ektron.Cms.API.Localization.GetFiles 706
Ektron.Cms.API.Localization.GetFilesInZipFile 707
Ektron.Cms.API.Localization.GetFlagUrl 708
Ektron.Cms.API.Localization.GetInterchangeFile 709
Ektron.Cms.API.Localization.GetInterchangeFiles 709
Ektron.Cms.API.Localization.GetInterchangeFilesInZipFile 709
Ektron.Cms.API.Localization.GetJobs 710
Ektron.Cms.API.Localization.GetLocalizationUrl 711
Ektron.Cms.API.Localization.GetTranslationUploadDirectory 711
Ektron.Cms.API.Localization.StartExportContentForTranslation 711
Ektron.Cms.API.Localization.StartExportFolderForTranslation 712
Ektron.Cms.API.Localization.StartExportMenusForTranslation 712
Ektron.Cms.API.Localization.StartImportTranslation 713
Ektron.Cms.API.Messaging 713
Ektron.Cms.API.Messaging.GetUnreadMessageCountForUser 713
Ektron.Cms.API.Metadata 714
Ektron.Cms.API.Metadata.AddMetaDataType 715
Ektron.Cms.API.Metadata.DeleteMetadataType 717
Ektron.Cms.API.Metadata.GetContentMetadataList 717, 719
Ektron.Cms.API.Metadata.GetIntegratedSearchFolder 719
Ektron.Cms.API.Metadata.GetIntegratedSearchFolders 719

Ektron.Cms.API.Metadata.GetMetadataByFolder 720
Ektron.Cms.API.Metadata.GetMetadataType 720
Ektron.Cms.API.Metadata.GetMetaDataTypes 720
Ektron.Cms.API.Metadata.UpdateMetadataType 721
Ektron.Cms.API.Permissions 721
Ektron.Cms.API.Permissions.AddItemPermission 722
Ektron.Cms.API.Permissions.DeleteItemPermission 722
Ektron.Cms.API.Permissions.DisableContentInheritance 722
Ektron.Cms.API.Permissions.DisableFolderInheritance 723
Ektron.Cms.API.Permissions.EnableContentInheritance 723
Ektron.Cms.API.Permissions.EnableFolderInheritance 723
Ektron.Cms.API.Permissions.GetUserPermissions 724
Ektron.Cms.API.Permissions.LoadPermissions 725
Ektron.Cms.API.Permissions.UpdateItemPermission 725
Ektron.Cms.API.Search 899
Ektron.Cms.API.Search namespace 899
Classes 899
Ektron.Cms.API.Search.SearchManager 899
Ektron.Cms.API.Search.SearchManager.Execute 900, 901
Ektron.Cms.API.Search.SearchManager.Search 903, 904, 905, 906
Ektron.Cms.API.Search.SearchManager.SearchForCategories 906, 907
Ektron.Cms.API.Search.SearchManager.SearchTable 908
Ektron.Cms.API.Site 725
Ektron.Cms.API.Site.BreakTaskPermissionInherit 726
Ektron.Cms.API.Site.DeleteCookie 727
Ektron.Cms.API.Site.GetAllActiveLanguages 727
Ektron.Cms.API.Site.GetAllLanguages 727, 728
Ektron.Cms.API.Site.GetClientDayNames 728
Ektron.Cms.API.Site.GetClientMonthNames 728
Ektron.Cms.API.Site.GetCookieValue 728
Ektron.Cms.API.Site.GetLanguage 729
Ektron.Cms.API.Site.GetLanguages 729
Ektron.Cms.API.Site.GetSiteVariables 729
Ektron.Cms.API.Site.GetTaskPermission 730
Ektron.Cms.API.Site.IsTaskPermissionInherited 730
Ektron.Cms.API.Site.SetCookieValue 730
Ektron.Cms.API.Site.SetTaskPermission 731
Ektron.Cms.API.Site.UpdateSiteEnabledLanguages 731
Ektron.Cms.API.Site.UpdateSiteVariables 731
Ektron.Cms.API.SiteMap 732
Ektron.Cms.API.SiteMap.GetFolderBreadcrumbPath 732
Ektron.Cms.API.SiteMap.GetWebsiteMap 734
Ektron.Cms.API.SiteMap.RenderFolderBreadcrumbHtml 735
Ektron.Cms.API.SiteMap.RenderWebsiteMap 736
Ektron.Cms.API.Subscription 736
Ektron.Cms.API.Subscription.AddEmailFrom 737
Ektron.Cms.API.Subscription.AddSubscription 738
Ektron.Cms.API.Subscription.AddSubscriptionMessage 738
Ektron.Cms.API.Subscription.DeleteEmailFrom 738
Ektron.Cms.API.Subscription.DeleteSubscription 738
Ektron.Cms.API.Subscription.DeleteSubscriptionMessage 739
Ektron.Cms.API.Subscription.GetAllActiveSubscriptions 739
Ektron.Cms.API.Subscription.GetAllEmailFrom 739
Ektron.Cms.API.Subscription.GetAllSubscriptionMessages 739
Ektron.Cms.API.Subscription.GetAllSubscriptions 740
Ektron.Cms.API.Subscription.GetFolderInheritedFrom 740
Ektron.Cms.API.Subscription.GetSubscription 740
Ektron.Cms.API.Subscription.GetSubscriptionMessage 740
Ektron.Cms.API.Subscription.GetSubscriptionMessagesForType 741
Ektron.Cms.API.Subscription.GetSubscriptionPropertiesForContent 741
Ektron.Cms.API.Subscription.GetSubscriptionPropertiesForFolder 741
Ektron.Cms.API.Subscription.GetSubscriptionsForContent 742
Ektron.Cms.API.Subscription.GetSubscriptionsForFolder 742
Ektron.Cms.API.Subscription.SendWebAlert 742
Ektron.Cms.API.Subscription.UpdateEmailFrom 743
Ektron.Cms.API.Subscription.UpdateSubscription 743
Ektron.Cms.API.Subscription.UpdateSubscriptionMessage 743
Ektron.Cms.API.Subscription.UpdateSubscriptionPropertiesForContent 743
Ektron.Cms.API.Subscription.UpdateSubscriptionPropertiesForFolder 744
Ektron.Cms.API.Task 909
Ektron.Cms.API.Task namespace 909

Classes 909
Ektron.Cms.API.Task.Task 909
Ektron.Cms.API.Task.Task.AddTask 910
Ektron.Cms.API.Task.Task.AssignTaskToUser 910, 911
Ektron.Cms.API.Task.Task.AssignTaskToUsergroup 911, 912
Ektron.Cms.API.Task.Task.DeleteTask 912, 913
Ektron.Cms.API.Task.Task.DeleteTasks 913
Ektron.Cms.API.Task.Task.GetTask 914
Ektron.Cms.API.Task.Task.GetTasks 914, 915
Ektron.Cms.API.Task.Task.GetTasksAssignedByUser 915
Ektron.Cms.API.Task.Task.GetTasksAssignedToUser 916, 917
Ektron.Cms.API.Task.Task.GetTasksAssignedToUsergroup 918
Ektron.Cms.API.Task.Task.GetTaskState 918
Ektron.Cms.API.Task.Task.SetStateForTask 919
Ektron.Cms.API.Task.Task.UpdateTask 919
Ektron.Cms.API.Task.TaskCategory 919
Ektron.Cms.API.Task.TaskCategory.AddTaskCategory 920
Ektron.Cms.API.Task.TaskCategory.DeleteTaskCategory 920
Ektron.Cms.API.Task.TaskCategory.GetAllCategories 921
Ektron.Cms.API.Task.TaskCategory.GetTaskCategory 921
Ektron.Cms.API.Task.TaskCategory.UpdateTaskCategory 921
Ektron.Cms.API.Task.TaskCategoryType 922
Ektron.Cms.API.Task.TaskCategoryType.AddTaskType 922
Ektron.Cms.API.Task.TaskCategoryType.DeleteTaskType 923
Ektron.Cms.API.Task.TaskCategoryType.GetTaskType 923
Ektron.Cms.API.Task.TaskCategoryType.UpdateTaskType 923
Ektron.Cms.API.UrlAliasing 924
Ektron.Cms.API.UrlAliasing namespace 924
Classes 924
Ektron.Cms.API.UrlAliasing.UrlAliasAuto 924
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.Add 925
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.ClearCache 925
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.Delete 926
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetAutoAliasNameTypes 926
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetAutoAliasTypes 926
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetFolderItem 926
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetFolderList 927
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetFolderSource 927
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetItem 927
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetItemForContent 928
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetList 928
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetListForContent 929
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetTarget 930
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetTaxonomyItem 930
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetTaxonomyList 931
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetTaxonomySource 931
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.Update 931
Ektron.Cms.API.UrlAliasing.UrlAliasCommon 931
Ektron.Cms.API.UrlAliasing.UrlAliasCommon.GetAliasForContent 932
Ektron.Cms.API.UrlAliasing.UrlAliasCommon.GetTarget 932, 933
Ektron.Cms.API.UrlAliasing.UrlAliasManual 933
Ektron.Cms.API.UrlAliasing.UrlAliasManual.Add 934
Ektron.Cms.API.UrlAliasing.UrlAliasManual.ClearCache 934
Ektron.Cms.API.UrlAliasing.UrlAliasManual.Delete 934
Ektron.Cms.API.UrlAliasing.UrlAliasManual.GetDefaultAlias 934
Ektron.Cms.API.UrlAliasing.UrlAliasManual.GetFileExtensions 935
Ektron.Cms.API.UrlAliasing.UrlAliasManual.GetItem 935
Ektron.Cms.API.UrlAliasing.UrlAliasManual.GetList 936, 937, 938, 939
Ektron.Cms.API.UrlAliasing.UrlAliasManual.GetSiteList 940
Ektron.Cms.API.UrlAliasing.UrlAliasManual.GetTemplateList 940
Ektron.Cms.API.UrlAliasing.UrlAliasManual.Update 940
Ektron.Cms.API.UrlAliasing.UrlAliasManual.ValidateName 940
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx 941
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.Add 941
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.ClearCache 942
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.Delete 942
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.EvaluateExpression 942
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.GetItem 942
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.GetList 943
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.GetPriorityList 943

Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.GetTarget 944
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.Update 944
Ektron.Cms.API.UrlAliasing.UrlAliasSettings 944
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.AutoCacheDuration 946
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.AutoCacheSize 946
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.DefaultPage 947
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.DisableLanguageAwareness 947
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnableAutoAliasing 947
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnableAutoCaching 947
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnabledTracing 947
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnableManualAliasing 948
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnableManualCaching 948
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnableRegExAliasing 948
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnableRegExCaching 948
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.IsAutoAliasEnabled 949
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.IsManualAliasEnabled 949
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.IsRegExAliasEnabled 949
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.ManualCacheDuration 949
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.ManualCacheSize 949
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.OverrideTemplate 950
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.PageExtensions 950
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.QueryStringAction 950
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.RefreshSettingsCache 945
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.RegExCacheDuration 950
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.RegExCacheSize 950
Ektron.Cms.API.User 950
Ektron.Cms.API.User namespace 950
Classes 951
Ektron.Cms.API.User.ADUser 951
Ektron.Cms.API.User.ADUser.AddADGroupToCMS 952
Ektron.Cms.API.User.ADUser.AddADUsersToCMSByUsername 952
Ektron.Cms.API.User.ADUser.DeSynchUserGroups 952
Ektron.Cms.API.User.ADUser.DeSynchUsers 953
Ektron.Cms.API.User.ADUser.GetADMapping 953
Ektron.Cms.API.User.ADUser.GetADStatus 953
Ektron.Cms.API.User.ADUser.GetAvailableADGroups 954
Ektron.Cms.API.User.ADUser.GetAvailableADUsers 954
Ektron.Cms.API.User.ADUser.GetCMSGroupsToSync 954
Ektron.Cms.API.User.ADUser.GetCMSRelationshipsToSync 955
Ektron.Cms.API.User.ADUser.GetCMSUsersToSync 955
Ektron.Cms.API.User.ADUser.GetDomains 955
Ektron.Cms.API.User.ADUser.MapCMSUserGroupToAD 956
Ektron.Cms.API.User.ADUser.MapCMSUserToAD 956
Ektron.Cms.API.User.ADUser.SynchCMSGroupsToAD 956
Ektron.Cms.API.User.ADUser.SynchCMSRelationshipsToAD 957
Ektron.Cms.API.User.ADUser.SynchCMSUsersToAD 957
Ektron.Cms.API.User.ADUser.UpdateADMapping 957
Ektron.Cms.API.User.ADUser.UpdateADUsersGroups 958
Ektron.Cms.API.User.ADUser.UpdateUsersInfoFromAD 958
Ektron.Cms.API.User.User 958
Ektron.Cms.API.User.User.ActivateUserAccount 961
Ektron.Cms.API.User.User.AddMembershipUser 961
Ektron.Cms.API.User.User.AddNewUser 963
Ektron.Cms.API.User.User.AddRolePermission 964

Ektron.Cms.API.User.User.AddUser 964
Ektron.Cms.API.User.User.AddUserGroup 965
Ektron.Cms.API.User.User.AddUserToGroup 965
Ektron.Cms.API.User.User.AutoLogInUser 965
Ektron.Cms.API.User.User.CreateBlankUserObject 966
Ektron.Cms.API.User.User.DeleteUserByID 967
Ektron.Cms.API.User.User.DeleteUserByIds 967
Ektron.Cms.API.User.User.DeleteUserFromGroup 967
Ektron.Cms.API.User.User.DeleteUserGroup 967
Ektron.Cms.API.User.User.EditUserCustomProperties 968
Ektron.Cms.API.User.User.GetActiveUser 968
Ektron.Cms.API.User.User.GetADMapping 968
Ektron.Cms.API.User.User.GetAllCmsUsers 969
Ektron.Cms.API.User.User.GetAllCustomProperties 969
Ektron.Cms.API.User.User.GetAllUserByCustomProperty 970
Ektron.Cms.API.User.User.GetAllUserGroups 970, 971
Ektron.Cms.API.User.User.GetAllUsers 971, 972
Ektron.Cms.API.User.User.GetGroupsUserIsIn 972
Ektron.Cms.API.User.User.GetMembershipUser 972
Ektron.Cms.API.User.User.GetNonVerifyUser 973
Ektron.Cms.API.User.User.GetUser 973
Ektron.Cms.API.User.User.GetUserByUsername 974
Ektron.Cms.API.User.User.GetUserCustomProperties 975
Ektron.Cms.API.User.User.GetUserEmailInfo 976
Ektron.Cms.API.User.User.GetUserForFolderAdmin 976
Ektron.Cms.API.User.User.GetUserGroup 976
Ektron.Cms.API.User.User.GetUserGroupByName 976
Ektron.Cms.API.User.User.GetUserGroupForFolderAdmin 977
Ektron.Cms.API.User.User.GetUserPermissions 977
Ektron.Cms.API.User.User.GetUserPreference 978
Ektron.Cms.API.User.User.GetUsers 978, 979
Ektron.Cms.API.User.User.GetUsersForTask 979
Ektron.Cms.API.User.User.GetUsersNotInGroup 980
Ektron.Cms.API.User.User.IsAdmin 980
Ektron.Cms.API.User.User.IsAGroupMember 980
Ektron.Cms.API.User.User.IsARoleMember 982, 983
Ektron.Cms.API.User.User.IsARoleMemberForContent 984
Ektron.Cms.API.User.User.IsARoleMemberForContent_FolderUserAdmin 984
Ektron.Cms.API.User.User.IsARoleMemberForFolder 984
Ektron.Cms.API.User.User.IsARoleMemberForFolder_FolderUserAdmin 985
Ektron.Cms.API.User.User.IsARoleMemberForFolder_MetadataAdmin 985
Ektron.Cms.API.User.User.LockUserAccount 986
Ektron.Cms.API.User.User.LogInUser 986, 987, 988
Ektron.Cms.API.User.User.ReadCustomProperties 990
Ektron.Cms.API.User.User.ResetMembershipUserPassword 990, 991
Ektron.Cms.API.User.User.SetAuthenticationCookie 991
Ektron.Cms.API.User.User.UnsubscribeUser 991
Ektron.Cms.API.User.User.UpdateCustomPropertiesItemOrder 992
Ektron.Cms.API.User.User.UpdateUser 992
Ektron.Cms.API.User.User.UpDateUserGroup 993
Ektron.Cms.API.User.User.UpdateUserPreferences 993
Ektron.Cms.API.User.User.UserIdAuthenticated 995
Ektron.Cms.API.User.User.UserObject 994
Ektron.Cms.API.Utilities 744
Ektron.Cms.API.Utilities.LoadPermissions 745
Ektron.Cms.AssetUpdateData 1606
Ektron.Cms.AssetUpdateData.AssetId 1607
Ektron.Cms.AssetUpdateData.Comment 1607
Ektron.Cms.AssetUpdateData.ContentId 1608
Ektron.Cms.AssetUpdateData.EndDate 1608
Ektron.Cms.AssetUpdateData.EndDateAction 1608
Ektron.Cms.AssetUpdateData.FileName 1609
Ektron.Cms.AssetUpdateData.FolderId 1609
Ektron.Cms.AssetUpdateData.GoLive 1609
Ektron.Cms.AssetUpdateData.LanguageId 1609
Ektron.Cms.AssetUpdateData.MetaData 1610
Ektron.Cms.AssetUpdateData.TaxonomyTreeIds 1610
Ektron.Cms.AssetUpdateData.Teaser 1610
Ektron.Cms.AssetUpdateData.Title 1610
Ektron.Cms.Commerce 1423
Ektron.Cms.Commerce namespace 1423
Classes 1426
Structs, Records, Enums 1581

Ektron.Cms.Commerce.AddressApi 602
Ektron.Cms.Commerce.AddressApi.Add 603
Ektron.Cms.Commerce.AddressApi.Delete 604
Ektron.Cms.Commerce.AddressApi.GetItem 604
Ektron.Cms.Commerce.AddressApi.GetList 604
Ektron.Cms.Commerce.AddressApi.Update 604
Ektron.Cms.Commerce.AddressApi.UpdateOrderAddress 605
Ektron.Cms.Commerce.AddressData 1428
Ektron.Cms.Commerce.AddressData.AddressLine1 1429
Ektron.Cms.Commerce.AddressData.AddressLine2 1429
Ektron.Cms.Commerce.AddressData.City 1429
Ektron.Cms.Commerce.AddressData.Company 1430
Ektron.Cms.Commerce.AddressData.Country 1430
Ektron.Cms.Commerce.AddressData.Id 1430
Ektron.Cms.Commerce.AddressData.Name 1430
Ektron.Cms.Commerce.AddressData.Phone 1430
Ektron.Cms.Commerce.AddressData.PostalCode 1430
Ektron.Cms.Commerce.AddressData.Region 1431
Ektron.Cms.Commerce.AddressProperty 1582
Ektron.Cms.Commerce.AddressProperty enumeration 1582
Ektron.Cms.Commerce.AdjustedBasketItem 1431
Ektron.Cms.Commerce.AdjustedBasketItem.AdjustedBasketItem
1431, 1432
Ektron.Cms.Commerce.AdjustedBasketItem.AdjustedTotal 1432
Ektron.Cms.Commerce.AdjustedBasketItem.ApplicableTaxRate
1432
Ektron.Cms.Commerce.AdjustedBasketItem.AppliedCoupons
1433
Ektron.Cms.Commerce.AdjustedBasketItem.HasFreeShipping
1433
Ektron.Cms.Commerce.AdjustedBasketItem.TaxesApplied 1433
Ektron.Cms.Commerce.AdjustedBasketItem.TotalCouponDiscount
1433
Ektron.Cms.Commerce.AppliedCouponData 1433
Ektron.Cms.Commerce.AppliedCouponData.AppliedCouponData
1434
Ektron.Cms.Commerce.AppliedCouponData.Coupon 1434
Ektron.Cms.Commerce.AppliedCouponData.CouponDiscount
1435
Ektron.Cms.Commerce.AppliedCouponDataCollection 1435
Ektron.Cms.Commerce.AppliedCouponDataCollection.FindByCouponId
1436
Ektron.Cms.Commerce.AppliedCouponDataCollection.ItemAdded
1435
Ektron.Cms.Commerce.AppliedCouponEventArgs 1436
Ektron.Cms.Commerce.AppliedCouponEventArgs.AppliedCoupon
1437
Ektron.Cms.Commerce.AppliedCouponEventArgs.AppliedCouponEventArgs
1436, 1437
Ektron.Cms.Commerce.Basket 1437
Ektron.Cms.Commerce.Basket.AddProduct 1438, 1439
Ektron.Cms.Commerce.Basket.AddProductVariant 1439
Ektron.Cms.Commerce.Basket.Decrement 1439
Ektron.Cms.Commerce.Basket.GetItemIndex 1440
Ektron.Cms.Commerce.Basket.GetKitConfiguration 1440
Ektron.Cms.Commerce.Basket.GetKitConfigurationAsAdmin
1440
Ektron.Cms.Commerce.Basket.IncrementItemQuantity 1440
Ektron.Cms.Commerce.Basket.MakeDefaultBasket 1440
Ektron.Cms.Commerce.Basket.RemoveItem 1441
Ektron.Cms.Commerce.Basket.RequestInformation 1442
Ektron.Cms.Commerce.Basket.SaveBasket 1441
Ektron.Cms.Commerce.Basket.UpdateItemQuantity 1441
Ektron.Cms.Commerce.Basket.UpdateKitConfiguration 1441
Ektron.Cms.Commerce.BasketApi 605
Ektron.Cms.Commerce.BasketApi.CalculateBasket 606, 607
Ektron.Cms.Commerce.BasketApi.CreateNewBasket 607
Ektron.Cms.Commerce.BasketApi.Delete 608
Ektron.Cms.Commerce.BasketApi.Empty 608
Ektron.Cms.Commerce.BasketApi.GetDefaultBasket 609
Ektron.Cms.Commerce.BasketApi.GetItem 609
Ektron.Cms.Commerce.BasketApi.GetList 610
Ektron.Cms.Commerce.BasketApi.SetDefaultBasket 610, 611
Ektron.Cms.Commerce.BasketCalculatorData 1442
Ektron.Cms.Commerce.BasketCalculatorData.AdjustedItems
1444
Ektron.Cms.Commerce.BasketCalculatorData.AllBasketCoupons
1444
Ektron.Cms.Commerce.BasketCalculatorData.Basket 1444

Ektron.Cms.Commerce.BasketCalculatorData.BasketCalculatorData 1443
Ektron.Cms.Commerce.BasketCalculatorData.BasketCoupons 1444
Ektron.Cms.Commerce.BasketCalculatorData.BasketTotal 1444
Ektron.Cms.Commerce.BasketCalculatorData.HasFreeShipping 1444
Ektron.Cms.Commerce.BasketCalculatorData.ShipFrom 1445
Ektron.Cms.Commerce.BasketCalculatorData.ShippingRates 1445
Ektron.Cms.Commerce.BasketCalculatorData.ShipTo 1445
Ektron.Cms.Commerce.BasketCalculatorData.TaxClassSubtotals 1445
Ektron.Cms.Commerce.BasketCalculatorData.TotalCouponDiscount 1445
Ektron.Cms.Commerce.BasketCalculatorData.TotalShippingCost 1446
Ektron.Cms.Commerce.BasketCalculatorData.TotalTaxes 1446
Ektron.Cms.Commerce.BasketCalculatorData.ValidationResults 1446
Ektron.Cms.Commerce.BasketData 1446
Ektron.Cms.Commerce.BasketData.DateCreated 1447
Ektron.Cms.Commerce.BasketData.DateModified 1447
Ektron.Cms.Commerce.BasketData.Id 1447
Ektron.Cms.Commerce.BasketData.IsDefault 1447
Ektron.Cms.Commerce.BasketData.Items 1447
Ektron.Cms.Commerce.BasketData.Name 1448
Ektron.Cms.Commerce.BasketData.ShippingAddressId 1448
Ektron.Cms.Commerce.BasketData.ShippingMethodId 1448
Ektron.Cms.Commerce.BasketData.ShopperId 1448
Ektron.Cms.Commerce.BasketData.Subtotal 1448
Ektron.Cms.Commerce.BasketData.UserId 1449
Ektron.Cms.Commerce.BasketItem 1449
Ektron.Cms.Commerce.BasketItem.AdjustedTotal 1450
Ektron.Cms.Commerce.BasketItem.BasketItem 1449
Ektron.Cms.Commerce.BasketItem.Configuration 1450
Ektron.Cms.Commerce.BasketItem.Dimensions 1451
Ektron.Cms.Commerce.BasketItem.Id 1451
Ektron.Cms.Commerce.BasketItem.IsInventoryManagementDisabled 1451
Ektron.Cms.Commerce.BasketItem.IsTangible 1451
Ektron.Cms.Commerce.BasketItem.ListPrice 1451
Ektron.Cms.Commerce.BasketItem.Pricing 1452
Ektron.Cms.Commerce.BasketItem.ProductId 1452
Ektron.Cms.Commerce.BasketItem.ProductLanguage 1452
Ektron.Cms.Commerce.BasketItem.ProductSku 1452
Ektron.Cms.Commerce.BasketItem.ProductTitle 1452
Ektron.Cms.Commerce.BasketItem.ProductType 1453
Ektron.Cms.Commerce.BasketItem.Quantity 1453
Ektron.Cms.Commerce.BasketItem.Quicklink 1453
Ektron.Cms.Commerce.BasketItem.SalePrice 1453
Ektron.Cms.Commerce.BasketItem.TaxClassId 1453
Ektron.Cms.Commerce.BasketItem.Variant 1453
Ektron.Cms.Commerce.BasketItem.Weight 1454
Ektron.Cms.Commerce.BasketItemCollection 1454
Ektron.Cms.Commerce.BasketItemCollection.AddRange 1455
Ektron.Cms.Commerce.BasketItemCollection.BasketItemCollection 1454
Ektron.Cms.Commerce.BasketItemCollection.Find 1455
Ektron.Cms.Commerce.BasketItemCollection.FindAll 1455
Ektron.Cms.Commerce.BasketItemCollection.FindIndex 1455
Ektron.Cms.Commerce.BasketVariantData 1455
Ektron.Cms.Commerce.BasketVariantData.BasketVariantData 1456
Ektron.Cms.Commerce.BasketVariantData.Dimensions 1456
Ektron.Cms.Commerce.BasketVariantData.Id 1456
Ektron.Cms.Commerce.BasketVariantData.Language 1457
Ektron.Cms.Commerce.BasketVariantData.ListPrice 1457
Ektron.Cms.Commerce.BasketVariantData.Pricing 1457
Ektron.Cms.Commerce.BasketVariantData.SalePrice 1457
Ektron.Cms.Commerce.BasketVariantData.Sku 1457
Ektron.Cms.Commerce.BasketVariantData.TaxClassId 1458
Ektron.Cms.Commerce.BasketVariantData.Title 1458
Ektron.Cms.Commerce.BasketVariantData.Weight 1458
Ektron.Cms.Commerce.BundleData 1458
Ektron.Cms.Commerce.BundleData.BundleData 1458
Ektron.Cms.Commerce.BundleData.BundledItems 1459
Ektron.Cms.Commerce.CatalogData 1459

Ektron.Cms.Commerce.CatalogData.Id 1460
Ektron.Cms.Commerce.CatalogData.Name 1460
Ektron.Cms.Commerce.CatalogEntryApi 611
Ektron.Cms.Commerce.CatalogEntryApi.Add 612
Ektron.Cms.Commerce.CatalogEntryApi.AddAndCheckIn 612
Ektron.Cms.Commerce.CatalogEntryApi.AddAndPublish 612
Ektron.Cms.Commerce.CatalogEntryApi.AddAndSave 612
Ektron.Cms.Commerce.CatalogEntryApi.Checkout 613
Ektron.Cms.Commerce.CatalogEntryApi.Delete 613
Ektron.Cms.Commerce.CatalogEntryApi.GetAttributeEntryList 613
Ektron.Cms.Commerce.CatalogEntryApi.GetItem 614
Ektron.Cms.Commerce.CatalogEntryApi.GetItemEdit 614
Ektron.Cms.Commerce.CatalogEntryApi.GetList 615
Ektron.Cms.Commerce.CatalogEntryApi.GetTaxonomyEntryList 615
Ektron.Cms.Commerce.CatalogEntryApi.Restore 615
Ektron.Cms.Commerce.CatalogEntryApi.UndoCheckOut 615
Ektron.Cms.Commerce.CatalogEntryApi.Update 616
Ektron.Cms.Commerce.CatalogEntryApi.UpdateAndCheckIn 616
Ektron.Cms.Commerce.CatalogEntryApi.UpdateAndPublish 616
Ektron.Cms.Commerce.ContractApi 616
Ektron.Cms.Commerce.ContractApi.Add 617
Ektron.Cms.Commerce.ContractApi.Delete 617
Ektron.Cms.Commerce.ContractApi.GetItem 617
Ektron.Cms.Commerce.ContractApi.GetList 618
Ektron.Cms.Commerce.ContractApi.Update 618
Ektron.Cms.Commerce.ContractData 1460
Ektron.Cms.Commerce.ContractData.ContractData 1461
Ektron.Cms.Commerce.ContractData.Id 1461
Ektron.Cms.Commerce.ContractData.Name 1461
Ektron.Cms.Commerce.CountryApi 618
Ektron.Cms.Commerce.CountryApi.Add 619
Ektron.Cms.Commerce.CountryApi.CanDelete 619
Ektron.Cms.Commerce.CountryApi.Delete 619
Ektron.Cms.Commerce.CountryApi.GetItem 620
Ektron.Cms.Commerce.CountryApi.GetList 620
Ektron.Cms.Commerce.CountryApi.Update 620
Ektron.Cms.Commerce.CountryData 1461
Ektron.Cms.Commerce.CountryData.CountryData 1462
Ektron.Cms.Commerce.CountryData.Enabled 1463
Ektron.Cms.Commerce.CountryData.Id 1464
Ektron.Cms.Commerce.CountryData.LongIsoCode 1464
Ektron.Cms.Commerce.CountryData.Name 1464
Ektron.Cms.Commerce.CountryData.ShortIsoCode 1464
Ektron.Cms.Commerce.CountryData.ToString 1463
Ektron.Cms.Commerce.CountryData.Validate 1463
Ektron.Cms.Commerce.CountryProperty 1582
Ektron.Cms.Commerce.CountryProperty enumeration 1582
Ektron.Cms.Commerce.CountryTaxRateData 1464
Ektron.Cms.Commerce.CountryTaxRateData.CountryData 1466
Ektron.Cms.Commerce.CountryTaxRateData.CountryTaxRateData 1465
Ektron.Cms.Commerce.CouponApi 620
Ektron.Cms.Commerce.CouponApi.Add 621
Ektron.Cms.Commerce.CouponApi.AddCouponToCatalog 622
Ektron.Cms.Commerce.CouponApi.AddCouponToProduct 622
Ektron.Cms.Commerce.CouponApi.AddCouponToTaxonomy 622
Ektron.Cms.Commerce.CouponApi.ApplyCouponToBasket 623
Ektron.Cms.Commerce.CouponApi.CalculateBasketCoupons 623
Ektron.Cms.Commerce.CouponApi.Deactivate 624
Ektron.Cms.Commerce.CouponApi.Delete 624
Ektron.Cms.Commerce.CouponApi.DeleteCouponProducts 624
Ektron.Cms.Commerce.CouponApi.GetCatalogList 624
Ektron.Cms.Commerce.CouponApi.GetCouponsAppliedToBasket 625
Ektron.Cms.Commerce.CouponApi.GetItem 625
Ektron.Cms.Commerce.CouponApi.GetList 625
Ektron.Cms.Commerce.CouponApi.GetProductList 626
Ektron.Cms.Commerce.CouponApi.GetTaxonomyList 626
Ektron.Cms.Commerce.CouponApi.IsCouponAppliedToProduct 626
Ektron.Cms.Commerce.CouponApi.IsCouponUsedForBasket 627
Ektron.Cms.Commerce.CouponApi.IsCouponUsedForOrder 627
Ektron.Cms.Commerce.CouponApi.SaveCouponApplications 627

Ektron.Cms.Commerce.CouponApi.Update 628
Ektron.Cms.Commerce.CouponApi.ValidateCoupon 628
Ektron.Cms.Commerce.CouponData 1466
Ektron.Cms.Commerce.CouponData.Code 1467
Ektron.Cms.Commerce.CouponData.CouponData 1466
Ektron.Cms.Commerce.CouponData.CouponType 1467
Ektron.Cms.Commerce.CouponData.CurrencyId 1467
Ektron.Cms.Commerce.CouponData.Description 1468
Ektron.Cms.Commerce.CouponData.DiscountType 1468
Ektron.Cms.Commerce.CouponData.DiscountValue 1468
Ektron.Cms.Commerce.CouponData.ExpirationDate 1468
Ektron.Cms.Commerce.CouponData.Id 1468
Ektron.Cms.Commerce.CouponData.IsActive 1468
Ektron.Cms.Commerce.CouponData.IsCombinable 1469
Ektron.Cms.Commerce.CouponData.IsRedeemable 1469
Ektron.Cms.Commerce.CouponData.MaximumAmount 1469
Ektron.Cms.Commerce.CouponData.MaximumUses 1469
Ektron.Cms.Commerce.CouponData.MinimumAmount 1469
Ektron.Cms.Commerce.CouponData.OnePerCustomer 1469
Ektron.Cms.Commerce.CouponData.StartDate 1470
Ektron.Cms.Commerce.CouponEntryData 1470
Ektron.Cms.Commerce.CouponEntryData.CouponEntryData 1470, 1471
Ektron.Cms.Commerce.CouponEntryData.CouponId 1471
Ektron.Cms.Commerce.CouponEntryData.DataState 1471
Ektron.Cms.Commerce.CouponEntryData.IsIncluded 1472
Ektron.Cms.Commerce.CouponEntryData.ObjectId 1472
Ektron.Cms.Commerce.CouponEntryData.ObjectType 1472
Ektron.Cms.Commerce.CouponEntryData.Path 1472
Ektron.Cms.Commerce.CouponEntryData.Title 1472
Ektron.Cms.Commerce.CouponProperty 1582
Ektron.Cms.Commerce.CouponProperty enumeration 1582
Ektron.Cms.Commerce.CurrencyApi 629
Ektron.Cms.Commerce.CurrencyApi.Add 629
Ektron.Cms.Commerce.CurrencyApi.CanDelete 629
Ektron.Cms.Commerce.CurrencyApi.Delete 630
Ektron.Cms.Commerce.CurrencyApi.GetItem 630
Ektron.Cms.Commerce.CurrencyApi.GetList 630
Ektron.Cms.Commerce.CurrencyApi.Update 630
Ektron.Cms.Commerce.CurrencyData 1472
Ektron.Cms.Commerce.CurrencyData.AlphaIsoCode 1473
Ektron.Cms.Commerce.CurrencyData.CultureCode 1473
Ektron.Cms.Commerce.CurrencyData.CurrencySymbol 1473
Ektron.Cms.Commerce.CurrencyData.Enabled 1474
Ektron.Cms.Commerce.CurrencyData.Id 1474
Ektron.Cms.Commerce.CurrencyData.ISOCurrencySymbol 1474
Ektron.Cms.Commerce.CurrencyData.Name 1474
Ektron.Cms.Commerce.CurrencyPricingData 1474
Ektron.Cms.Commerce.CurrencyPricingData.AlphaIsoCode 1477
Ektron.Cms.Commerce.CurrencyPricingData.Clone 1476
Ektron.Cms.Commerce.CurrencyPricingData.CurrencyId 1477
Ektron.Cms.Commerce.CurrencyPricingData.CurrencyPricingDa ta
1475
Ektron.Cms.Commerce.CurrencyPricingData.GetSalePrice 1476
Ektron.Cms.Commerce.CurrencyPricingData.ListPrice 1477
Ektron.Cms.Commerce.CurrencyPricingData.NumericIsoCode 1477
Ektron.Cms.Commerce.CurrencyPricingData.PricingType 1478
Ektron.Cms.Commerce.CurrencyPricingData.SalePrice 1478
Ektron.Cms.Commerce.CurrencyPricingData.TierPrices 1478
Ektron.Cms.Commerce.CurrencyPricingData.Validate 1476
Ektron.Cms.Commerce.CurrencyProperty 1583
Ektron.Cms.Commerce.CurrencyProperty enumeration 1583
Ektron.Cms.Commerce.CustomerApi 631
Ektron.Cms.Commerce.CustomerApi.Add 632
Ektron.Cms.Commerce.CustomerApi.ChangeBillingAddress 632
Ektron.Cms.Commerce.CustomerApi.ChangeShippingAddress 633
Ektron.Cms.Commerce.CustomerApi.Delete 633
Ektron.Cms.Commerce.CustomerApi.GetItem 633
Ektron.Cms.Commerce.CustomerApi.GetList 633
Ektron.Cms.Commerce.CustomerApi.GetOrderList 634
Ektron.Cms.Commerce.CustomerApi.GetOrderTotal 634
Ektron.Cms.Commerce.CustomerApi.Update 634
Ektron.Cms.Commerce.CustomerData 1478
Ektron.Cms.Commerce.CustomerData.AverageOrderValue 1479
Ektron.Cms.Commerce.CustomerData.BillingAddressId 1479
Ektron.Cms.Commerce.CustomerData.DisplayName 1479

Ektron.Cms.Commerce.CustomerData.EmailAddress 1479
Ektron.Cms.Commerce.CustomerData.FirstName 1480
Ektron.Cms.Commerce.CustomerData.Id 1480
Ektron.Cms.Commerce.CustomerData.IsDeleted 1480
Ektron.Cms.Commerce.CustomerData.IsMembershipUser 1480
Ektron.Cms.Commerce.CustomerData.LastName 1480
Ektron.Cms.Commerce.CustomerData.Password 1480
Ektron.Cms.Commerce.CustomerData.ShippingAddressId 1481
Ektron.Cms.Commerce.CustomerData.TotalOrders 1481
Ektron.Cms.Commerce.CustomerData.TotalOrderValue 1481
Ektron.Cms.Commerce.CustomerData.UserId 1481
Ektron.Cms.Commerce.CustomerData.UserName 1481
Ektron.Cms.Commerce.CustomerProperty 1583
Ektron.Cms.Commerce.CustomerProperty enumeration 1583
Ektron.Cms.Commerce.Dimensions 1482
Ektron.Cms.Commerce.Dimensions.Clone 1483
Ektron.Cms.Commerce.Dimensions.CompareTo 1483
Ektron.Cms.Commerce.Dimensions.ConvertUnits 1483
Ektron.Cms.Commerce.Dimensions.CubicUnits 1484
Ektron.Cms.Commerce.Dimensions.Dimensions 1482
Ektron.Cms.Commerce.Dimensions.Height 1484
Ektron.Cms.Commerce.Dimensions.Length 1484
Ektron.Cms.Commerce.Dimensions.Units 1484
Ektron.Cms.Commerce.Dimensions.Width 1484
Ektron.Cms.Commerce.EntryAttributeData 1485
Ektron.Cms.Commerce.EntryAttributeData.AttributeTypeId 1487
Ektron.Cms.Commerce.EntryAttributeData.CurrentValue 1487
Ektron.Cms.Commerce.EntryAttributeData.EntryAttributeData 1485, 1486
Ektron.Cms.Commerce.EntryAttributeData.VersionId 1487
Ektron.Cms.Commerce.EntryAttributeProperty 1583
Ektron.Cms.Commerce.EntryAttributeProperty enumeration 1583
Ektron.Cms.Commerce.EntryData 1487
Ektron.Cms.Commerce.EntryData.Attributes 1490
Ektron.Cms.Commerce.EntryData.Comment 1490
Ektron.Cms.Commerce.EntryData.ContentStatus 1490
Ektron.Cms.Commerce.EntryData.CurrencyId 1490
Ektron.Cms.Commerce.EntryData.DateModified 1490
Ektron.Cms.Commerce.EntryData.Dimensions 1490
Ektron.Cms.Commerce.EntryData.DisableInventoryManagement 1491
Ektron.Cms.Commerce.EntryData.EndDate 1491
Ektron.Cms.Commerce.EntryData.EndDateAction 1491
Ektron.Cms.Commerce.EntryData.EntryData 1488
Ektron.Cms.Commerce.EntryData.EntryType 1491
Ektron.Cms.Commerce.EntryData.FolderId 1492
Ektron.Cms.Commerce.EntryData.GoLive 1492
Ektron.Cms.Commerce.EntryData.Html 1492
Ektron.Cms.Commerce.EntryData.Id 1492
Ektron.Cms.Commerce.EntryData.Image 1493
Ektron.Cms.Commerce.EntryData.ImageThumbnail 1493
Ektron.Cms.Commerce.EntryData.IsArchived 1493
Ektron.Cms.Commerce.EntryData.IsBuyable 1493
Ektron.Cms.Commerce.EntryData.IsMarkedForDeletion 1493
Ektron.Cms.Commerce.EntryData.IsPublished 1494
Ektron.Cms.Commerce.EntryData.IsTangible 1494
Ektron.Cms.Commerce.EntryData.LanguageId 1494
Ektron.Cms.Commerce.EntryData.LastEditorFirstName 1494
Ektron.Cms.Commerce.EntryData.LastEditorId 1494
Ektron.Cms.Commerce.EntryData.LastEditorLastName 1495
Ektron.Cms.Commerce.EntryData.ListPrice 1495
Ektron.Cms.Commerce.EntryData.Media 1495
Ektron.Cms.Commerce.EntryData.Metadata 1495
Ektron.Cms.Commerce.EntryData.NumberSold 1495
Ektron.Cms.Commerce.EntryData.Pricing 1496
Ektron.Cms.Commerce.EntryData.ProductType 1496
Ektron.Cms.Commerce.EntryData.QuantityMultiple 1496
Ektron.Cms.Commerce.EntryData.Quicklink 1496
Ektron.Cms.Commerce.EntryData.SalePrice 1497
Ektron.Cms.Commerce.EntryData.Sku 1497
Ektron.Cms.Commerce.EntryData.Status 1497
Ektron.Cms.Commerce.EntryData.StatusLanguage 1497
Ektron.Cms.Commerce.EntryData.Summary 1497
Ektron.Cms.Commerce.EntryData.TaxClassId 1498
Ektron.Cms.Commerce.EntryData.TemplateId 1498
Ektron.Cms.Commerce.EntryData.Title 1498
Ektron.Cms.Commerce.EntryData.Url 1498
Ektron.Cms.Commerce.EntryData.VersionId 1498

Ektron.Cms.Commerce.EntryData.Weight 1499

Ektron.Cms.Commerce.EntryGroupData 1499

Ektron.Cms.Commerce.EntryGroupData.Description 1500

Ektron.Cms.Commerce.EntryGroupData.DisplayOrder 1500

Ektron.Cms.Commerce.EntryGroupData.EntryGroupData 1499, 1500

Ektron.Cms.Commerce.EntryGroupData.EntryId 1501

Ektron.Cms.Commerce.EntryGroupData.GroupType 1501

Ektron.Cms.Commerce.EntryGroupData.Id 1501

Ektron.Cms.Commerce.EntryGroupData.Name 1501

Ektron.Cms.Commerce.EntryGroupItemData 1501

Ektron.Cms.Commerce.EntryGroupItemData.DisplayOrder 1502

Ektron.Cms.Commerce.EntryGroupItemData.EntryGroupItemData
1502

Ektron.Cms.Commerce.EntryGroupItemData.EntryId 1503

Ektron.Cms.Commerce.EntryGroupItemData.GroupId 1503

Ektron.Cms.Commerce.EntryGroupItemData.Id 1503

Ektron.Cms.Commerce.EntryGroupItemProperty 1583

Ektron.Cms.Commerce.EntryGroupItemProperty enumeration 1583

Ektron.Cms.Commerce.EntryGroupProperty 1584

Ektron.Cms.Commerce.EntryGroupProperty enumeration 1584

Ektron.Cms.Commerce.EntryProperty 1584

Ektron.Cms.Commerce.EntryProperty enumeration 1584

Ektron.Cms.Commerce.ExchangeRateApi 634

Ektron.Cms.Commerce.ExchangeRateApi.Add 635

Ektron.Cms.Commerce.ExchangeRateApi.ConvertPrice 635

Ektron.Cms.Commerce.ExchangeRateApi.Delete 636

Ektron.Cms.Commerce.ExchangeRateApi.GetCurrentExchangeRate
636

Ektron.Cms.Commerce.ExchangeRateApi.GetCurrentList 636

Ektron.Cms.Commerce.ExchangeRateApi.GetList 636

Ektron.Cms.Commerce.ExchangeRateApi.Update 637

Ektron.Cms.Commerce.ExchangeRateData 1503

Ektron.Cms.Commerce.ExchangeRateData.BaseCurrencyId
1505

Ektron.Cms.Commerce.ExchangeRateData.ConvertPrice 1504

Ektron.Cms.Commerce.ExchangeRateData.EffectiveDate 1505

Ektron.Cms.Commerce.ExchangeRateData.ExchangeCurrencyId
1505

Ektron.Cms.Commerce.ExchangeRateData.ExchangeRateData
1504

Ektron.Cms.Commerce.ExchangeRateData.Rate 1505

Ektron.Cms.Commerce.ExchangeRateData.Validate 1504

Ektron.Cms.Commerce.ExchangeRateProperty 1584

Ektron.Cms.Commerce.ExchangeRateProperty enumeration
1584

Ektron.Cms.Commerce.Inventory 1407

Ektron.Cms.Commerce.Inventory namespace 1407

Ektron.Cms.Commerce.Inventory.Provider 1407

Ektron.Cms.Commerce.Inventory.Provider namespace 1407

Ektron.Cms.Commerce.Inventory.Provider.InventoryProvider
1407

Ektron.Cms.Commerce.Inventory.Provider.InventoryProvider.DecreaseStockLevel
1408

Ektron.Cms.Commerce.Inventory.Provider.InventoryProvider.GetInventory
1409

Ektron.Cms.Commerce.Inventory.Provider.InventoryProvider.GetReorderLevel
1409

Ektron.Cms.Commerce.Inventory.Provider.InventoryProvider.GetUnitsInStock
1410

Ektron.Cms.Commerce.Inventory.Provider.InventoryProvider.GetUnitsOnOrder
1410

Ektron.Cms.Commerce.Inventory.Provider.InventoryProvider.IncreaseStockLevel
1410

Ektron.Cms.Commerce.Inventory.Provider.InventoryProvider.SaveInventory
1411

Ektron.Cms.Commerce.InventoryApi 637

Ektron.Cms.Commerce.InventoryApi.SaveInventory 637

Ektron.Cms.Commerce.InventoryData 1506

Ektron.Cms.Commerce.InventoryData.EntryId 1507

Ektron.Cms.Commerce.InventoryData.ReorderLevel 1507

Ektron.Cms.Commerce.InventoryData.UnitsInStock 1507

Ektron.Cms.Commerce.InventoryData.UnitsOnOrder 1507

Ektron.Cms.Commerce.KitConfigData 1508

Ektron.Cms.Commerce.KitConfigData.BasketItemId 1509

Ektron.Cms.Commerce.KitConfigData.GetGroupIndex 1508

Ektron.Cms.Commerce.KitConfigData.Groups 1509
Ektron.Cms.Commerce.KitConfigData.Id 1509
Ektron.Cms.Commerce.KitConfigData.KitConfigData 1508
Ektron.Cms.Commerce.KitConfigData.PriceModifier 1509
Ektron.Cms.Commerce.KitData 1510
Ektron.Cms.Commerce.KitData.KitData 1510
Ektron.Cms.Commerce.KitData.OptionGroups 1510
Ektron.Cms.Commerce.LinearUnit 1585
Ektron.Cms.Commerce.LinearUnit enumeration 1585
Ektron.Cms.Commerce.OptionGroupData 1514
Ektron.Cms.Commerce.OptionGroupData.Image 1516
Ektron.Cms.Commerce.OptionGroupData.OptionGroupData 1515
Ektron.Cms.Commerce.OptionGroupData.Options 1516
Ektron.Cms.Commerce.OptionGroupItemCollection 1516
Ektron.Cms.Commerce.OptionGroupItemCollection.AddRange 1517
Ektron.Cms.Commerce.OptionGroupItemCollection.Find 1517
Ektron.Cms.Commerce.OptionGroupItemCollection.FindAll 1517
Ektron.Cms.Commerce.OptionGroupItemCollection.FindIndex 1517
Ektron.Cms.Commerce.OptionGroupItemCollection.OptionGroupItemCollection 1516
Ektron.Cms.Commerce.OptionGroupItemData 1518
Ektron.Cms.Commerce.OptionGroupItemData.CurrencyId 1519
Ektron.Cms.Commerce.OptionGroupItemData.ExtraText 1519
Ektron.Cms.Commerce.OptionGroupItemData.Name 1520
Ektron.Cms.Commerce.OptionGroupItemData.OptionGroupItemData 1518, 1519
Ektron.Cms.Commerce.OptionGroupItemData.PriceModification 1520
Ektron.Cms.Commerce.OptionGroupItemData.PriceModificationType 1520
Ektron.Cms.Commerce.OrderApi 638
Ektron.Cms.Commerce.OrderApi.Capture 639
Ektron.Cms.Commerce.OrderApi.GetCustomerOrderList 639
Ektron.Cms.Commerce.OrderApi.GetItem 640
Ektron.Cms.Commerce.OrderApi.GetList 640, 641
Ektron.Cms.Commerce.OrderApi.GetOnHoldOrderList 641
Ektron.Cms.Commerce.OrderApi.GetOrderMessage 642
Ektron.Cms.Commerce.OrderApi.GetOrderPayment 643
Ektron.Cms.Commerce.OrderApi.GetOrderStatus 643
Ektron.Cms.Commerce.OrderApi.GetProductOrderList 643, 644
Ektron.Cms.Commerce.OrderApi.GetReport 644
Ektron.Cms.Commerce.OrderApi.ItemsInStock 644
Ektron.Cms.Commerce.OrderApi.MarkAsFraud 645
Ektron.Cms.Commerce.OrderApi.MarkPaymentAsSettled 645
Ektron.Cms.Commerce.OrderApi.ProcessOrderInventory 646
Ektron.Cms.Commerce.OrderApi.RollbackOrderInventory 646
Ektron.Cms.Commerce.OrderApi.SetOrderStatus 646
Ektron.Cms.Commerce.OrderApi.SetOrderToHold 646
Ektron.Cms.Commerce.OrderApi.SetTrackingNumber 647
Ektron.Cms.Commerce.OrderData 1520
Ektron.Cms.Commerce.OrderData.BillingAddressId 1521
Ektron.Cms.Commerce.OrderData.ClientIP 1521
Ektron.Cms.Commerce.OrderData.CouponTotal 1521
Ektron.Cms.Commerce.OrderData.CouponUsed 1522
Ektron.Cms.Commerce.OrderData.Currency 1522
Ektron.Cms.Commerce.OrderData.Customer 1522
Ektron.Cms.Commerce.OrderData.CustomerType 1522
Ektron.Cms.Commerce.OrderData.DateCompleted 1522
Ektron.Cms.Commerce.OrderData.DateCreated 1523
Ektron.Cms.Commerce.OrderData.DateRequired 1523
Ektron.Cms.Commerce.OrderData.HandlingTotal 1523
Ektron.Cms.Commerce.OrderData.HttpHost 1523
Ektron.Cms.Commerce.OrderData.Id 1523
Ektron.Cms.Commerce.OrderData.OrderTotal 1524
Ektron.Cms.Commerce.OrderData.Parts 1524
Ektron.Cms.Commerce.OrderData.ShippingTotal 1524
Ektron.Cms.Commerce.OrderData.SpecialInstructions 1524
Ektron.Cms.Commerce.OrderData.StageId 1524
Ektron.Cms.Commerce.OrderData.StageName 1525
Ektron.Cms.Commerce.OrderData.Status 1525
Ektron.Cms.Commerce.OrderData.Subtotal 1525
Ektron.Cms.Commerce.OrderData.TaxTotal 1525
Ektron.Cms.Commerce.OrderData.VisitorId 1525
Ektron.Cms.Commerce.OrderLineData 1526
Ektron.Cms.Commerce.OrderLineData.Configuration 1526
Ektron.Cms.Commerce.OrderLineData.ItemType 1526

Ektron.Cms.Commerce.OrderLineData.LineItemId 1527
Ektron.Cms.Commerce.OrderLineData.OrderPartId 1527
Ektron.Cms.Commerce.OrderLineData.PriceEach 1527
Ektron.Cms.Commerce.OrderLineData.PriceSubTotal 1527
Ektron.Cms.Commerce.OrderLineData.PriceTotal 1527
Ektron.Cms.Commerce.OrderLineData.ProductId 1528
Ektron.Cms.Commerce.OrderLineData.ProductLanguage 1528
Ektron.Cms.Commerce.OrderLineData.Quantity 1528
Ektron.Cms.Commerce.OrderPartData 1528
Ektron.Cms.Commerce.OrderPartData.DateShipped 1529
Ektron.Cms.Commerce.OrderPartData.HandlingTotal 1529
Ektron.Cms.Commerce.OrderPartData.Lines 1529
Ektron.Cms.Commerce.OrderPartData.OrderId 1530
Ektron.Cms.Commerce.OrderPartData.PartId 1530
Ektron.Cms.Commerce.OrderPartData.PartTotal 1530
Ektron.Cms.Commerce.OrderPartData.ShippingAddressId 1530
Ektron.Cms.Commerce.OrderPartData.ShippingTotal 1530
Ektron.Cms.Commerce.OrderPartData.Status 1531
Ektron.Cms.Commerce.OrderPartData.Subtotal 1531
Ektron.Cms.Commerce.OrderPartData.SupplierId 1531
Ektron.Cms.Commerce.OrderPartData.TaxTotal 1531
Ektron.Cms.Commerce.OrderPartData.TrackingNumber 1531
Ektron.Cms.Commerce.OrderProperty 1585
Ektron.Cms.Commerce.OrderProperty enumeration 1585
Ektron.Cms.Commerce.PackageApi 647
Ektron.Cms.Commerce.PackageApi.Add 647
Ektron.Cms.Commerce.PackageApi.Delete 648
Ektron.Cms.Commerce.PackageApi.GetItem 648
Ektron.Cms.Commerce.PackageApi.GetList 648
Ektron.Cms.Commerce.PackageApi.Update 649
Ektron.Cms.Commerce.PackageData 1532
Ektron.Cms.Commerce.PackageData.Dimensions 1533
Ektron.Cms.Commerce.PackageData.Id 1533
Ektron.Cms.Commerce.PackageData.MaxWeight 1533
Ektron.Cms.Commerce.PackageData.Name 1534
Ektron.Cms.Commerce.PackageData.PackageData 1532, 1533
Ektron.Cms.Commerce.PackageProperty 1585
Ektron.Cms.Commerce.PackageProperty enumeration 1585
Ektron.Cms.Commerce.PaymentAuthorization 1534
Ektron.Cms.Commerce.PaymentAuthorization.AuthorizedOn 1534
Ektron.Cms.Commerce.PaymentAuthorization.CapturedOn 1535
Ektron.Cms.Commerce.PaymentAuthorization.CreditedOn 1535
Ektron.Cms.Commerce.PaymentAuthorization.Gateway 1535
Ektron.Cms.Commerce.PaymentAuthorization.Message 1535
Ektron.Cms.Commerce.PaymentAuthorization.RecurringTransactionId 1535
Ektron.Cms.Commerce.PaymentAuthorization.TransactionId 1535
Ektron.Cms.Commerce.PaymentAuthorization.VoidedOn 1536
Ektron.Cms.Commerce.PaymentGatewayProvider 1412
Ektron.Cms.Commerce.PaymentGatewayProvider.Amount 1415
Ektron.Cms.Commerce.PaymentGatewayProvider.Authorization 1416
Ektron.Cms.Commerce.PaymentGatewayProvider.Authorize 1412
Ektron.Cms.Commerce.PaymentGatewayProvider.AuthorizeAndCapture 1413
Ektron.Cms.Commerce.PaymentGatewayProvider.AuthorizeRecurringPayment 1413
Ektron.Cms.Commerce.PaymentGatewayProvider.CancelRecurringPayment 1413
Ektron.Cms.Commerce.PaymentGatewayProvider.CapturePreauthorization 1414
Ektron.Cms.Commerce.PaymentGatewayProvider.Config 1416
Ektron.Cms.Commerce.PaymentGatewayProvider.CurrencyCode 1416
Ektron.Cms.Commerce.PaymentGatewayProvider.CurrencyId 1416
Ektron.Cms.Commerce.PaymentGatewayProvider.GatewayData 1416
Ektron.Cms.Commerce.PaymentGatewayProvider.InTestMode 1416
Ektron.Cms.Commerce.PaymentGatewayProvider.IsSubmissionSuccess 1417
Ektron.Cms.Commerce.PaymentGatewayProvider.OrderId 1417
Ektron.Cms.Commerce.PaymentGatewayProvider.PaymentMet

hod
1417

Ektron.Cms.Commerce.PaymentGatewayProvider.RecurringPaymentFailed
1414

Ektron.Cms.Commerce.PaymentGatewayProvider.RecurringPaymentProcessed
1414

Ektron.Cms.Commerce.PaymentGatewayProvider.SubmissionError
1417

Ektron.Cms.Commerce.PaymentGatewayProvider.SupportsCheckPayments
1417

Ektron.Cms.Commerce.PaymentGatewayProvider.SupportsCreditCardPayments
1418

Ektron.Cms.Commerce.PaymentGatewayProvider.SupportsRecurringPayments
1418

Ektron.Cms.Commerce.PaymentGatewayProvider.VoidPreAuthorization
1414

Ektron.Cms.Commerce.PaymentMethod 1536

Ektron.Cms.Commerce.PaymentMethod.BillingAddress 1537

Ektron.Cms.Commerce.PaymentMethod.BillingFirstName 1537

Ektron.Cms.Commerce.PaymentMethod.BillingLastName 1537

Ektron.Cms.Commerce.PaymentMethod.GetPayment 1536

Ektron.Cms.Commerce.PipelineStageData 1537

Ektron.Cms.Commerce.PipelineStageData.Name 1538

Ektron.Cms.Commerce.PipelineStageData.Sequence 1538

Ektron.Cms.Commerce.PipelineStageData.StatusId 1538

Ektron.Cms.Commerce.PostalCodeTaxRateData 1539

Ektron.Cms.Commerce.PostalCodeTaxRateData.PostalCode
1540

Ektron.Cms.Commerce.PostalCodeTaxRateData.PostalCodeTaxRateData
1539, 1540

Ektron.Cms.Commerce.PostalCodeTaxRateData.RegionId 1540

Ektron.Cms.Commerce.PricingData 1540

Ektron.Cms.Commerce.PricingData.CurrencyPricelist 1545

Ektron.Cms.Commerce.PricingData.GetCurrencyById 1542

Ektron.Cms.Commerce.PricingData.GetCurrencyByIso 1542

Ektron.Cms.Commerce.PricingData.GetCurrencyByIsoNumber
1543

Ektron.Cms.Commerce.PricingData.GetListPrice 1543

Ektron.Cms.Commerce.PricingData.GetListPriceByCurrencyId
1543

Ektron.Cms.Commerce.PricingData.GetListPriceByIsoNumber
1543

Ektron.Cms.Commerce.PricingData.GetSalePrice 1543

Ektron.Cms.Commerce.PricingData.GetSalePriceByCurrencyId
1544

Ektron.Cms.Commerce.PricingData.GetSalePriceByIsoNumber
1544

Ektron.Cms.Commerce.PricingData.IsRecurringPrice 1545

Ektron.Cms.Commerce.PricingData.PricingData 1541

Ektron.Cms.Commerce.PricingData.Recurrence 1545

Ektron.Cms.Commerce.PricingData.Validate 1544

Ektron.Cms.Commerce.ProductCouponEntryData 1574

Ektron.Cms.Commerce.ProductCouponEntryData.EntryType
1575

Ektron.Cms.Commerce.ProductCouponEntryData.ProductCouponEntryData
1575

Ektron.Cms.Commerce.ProductData 1545

Ektron.Cms.Commerce.ProductData.ProductData 1546

Ektron.Cms.Commerce.ProductData.Variants 1546

Ektron.Cms.Commerce.ProductTypeApi 649

Ektron.Cms.Commerce.ProductTypeApi.Add 650

Ektron.Cms.Commerce.ProductTypeApi.AddProductAttribute
650

Ektron.Cms.Commerce.ProductTypeApi.Delete 650

Ektron.Cms.Commerce.ProductTypeApi.DeleteProductAttribute
650, 651

Ektron.Cms.Commerce.ProductTypeApi.GetFolderProductTypeList
651

Ektron.Cms.Commerce.ProductTypeApi.GetItem 651, 652

Ektron.Cms.Commerce.ProductTypeApi.GetProductAttribute
652

Ektron.Cms.Commerce.ProductTypeApi.GetProductAttributeList
652

Ektron.Cms.Commerce.ProductTypeApi.Update 652

Ektron.Cms.Commerce.ProductTypeApi.UpdateProductAttribute
652

Ektron.Cms.Commerce.ProductTypeAttributeData 1546

Ektron.Cms.Commerce.ProductTypeAttributeData.ActiveStatus
1548

Ektron.Cms.Commerce.ProductTypeAttributeData.DataType 1548

Ektron.Cms.Commerce.ProductTypeAttributeData.DefaultValue 1548

Ektron.Cms.Commerce.ProductTypeAttributeData.DisplayOrder 1548

Ektron.Cms.Commerce.ProductTypeAttributeData.Id 1549

Ektron.Cms.Commerce.ProductTypeAttributeData.Name 1549

Ektron.Cms.Commerce.ProductTypeAttributeData.ProductTypeAttributeData 1547

Ektron.Cms.Commerce.ProductTypeAttributeProperty 1586

Ektron.Cms.Commerce.ProductTypeAttributeProperty enumeration 1586

Ektron.Cms.Commerce.ProductTypeData 1549

Ektron.Cms.Commerce.ProductTypeData.Attributes 1550

Ektron.Cms.Commerce.ProductTypeData.DefaultThumbnails 1550

Ektron.Cms.Commerce.ProductTypeData.EntryClass 1550

Ektron.Cms.Commerce.ProductTypeProperty 1586

Ektron.Cms.Commerce.ProductTypeProperty enumeration 1586

Ektron.Cms.Commerce.RecommendationApi 653

Ektron.Cms.Commerce.RecommendationApi.Add 653

Ektron.Cms.Commerce.RecommendationApi.Delete 653

Ektron.Cms.Commerce.RecommendationApi.DeleteRecommendationByEntry 654

Ektron.Cms.Commerce.RecommendationApi.GetItem 654

Ektron.Cms.Commerce.RecommendationApi.Update 654

Ektron.Cms.Commerce.RecommendationApi.UpdateCrossSell 654

Ektron.Cms.Commerce.RecommendationApi.UpdateUpSell 655

Ektron.Cms.Commerce.RecommendationItemData 1550

Ektron.Cms.Commerce.RecommendationItemData.EntryLanguage 1552

Ektron.Cms.Commerce.RecommendationItemData.EntryType 1552

Ektron.Cms.Commerce.RecommendationItemData.Html 1552

Ektron.Cms.Commerce.RecommendationItemData.Image 1553

Ektron.Cms.Commerce.RecommendationItemData.ImageThumbnail 1553

Ektron.Cms.Commerce.RecommendationItemData.Pricing 1553

Ektron.Cms.Commerce.RecommendationItemData.Quicklink 1553

Ektron.Cms.Commerce.RecommendationItemData.RecommendationItemData 1551

Ektron.Cms.Commerce.RecommendationItemData.Summary 1553

Ektron.Cms.Commerce.RecommendationItemData.Title 1553

Ektron.Cms.Commerce.RecommendationItemData.Type 1554

Ektron.Cms.Commerce.RegionApi 655

Ektron.Cms.Commerce.RegionApi.Add 655

Ektron.Cms.Commerce.RegionApi.CanDelete 656

Ektron.Cms.Commerce.RegionApi.Delete 656

Ektron.Cms.Commerce.RegionApi.GetItem 656

Ektron.Cms.Commerce.RegionApi.GetList 657

Ektron.Cms.Commerce.RegionApi.Update 657

Ektron.Cms.Commerce.RegionData 1554

Ektron.Cms.Commerce.RegionData.Code 1556

Ektron.Cms.Commerce.RegionData.CountryId 1556

Ektron.Cms.Commerce.RegionData.Enabled 1556

Ektron.Cms.Commerce.RegionData.Id 1557

Ektron.Cms.Commerce.RegionData.Name 1557

Ektron.Cms.Commerce.RegionData.RegionData 1555

Ektron.Cms.Commerce.RegionData.ToString 1555

Ektron.Cms.Commerce.RegionData.Validate 1555

Ektron.Cms.Commerce.RegionProperty 1586

Ektron.Cms.Commerce.RegionProperty enumeration 1586

Ektron.Cms.Commerce.RegionTaxRateData 1557

Ektron.Cms.Commerce.RegionTaxRateData.RegionData 1558

Ektron.Cms.Commerce.RegionTaxRateData.RegionTaxRateData 1558

Ektron.Cms.Commerce.Shipment 1418

Ektron.Cms.Commerce.Shipment namespace 1418

Ektron.Cms.Commerce.Shipment.Provider 1418

Ektron.Cms.Commerce.Shipment.Provider namespace 1418

Ektron.Cms.Commerce.Shipment.Provider.ShipmentProvider 1419

Ektron.Cms.Commerce.Shipment.Provider.ShipmentProvider.AccountNumber 1421

Ektron.Cms.Commerce.Shipment.Provider.ShipmentProvider.GetRates 1419

Ektron.Cms.Commerce.Shipment.Provider.ShipmentProvider.GetServiceTypes 1420
Ektron.Cms.Commerce.Shipment.Provider.ShipmentProvider.GetTrackingUrl 1420
Ektron.Cms.Commerce.Shipment.Provider.ShipmentProvider.IsTrackingSupported 1421
Ektron.Cms.Commerce.Shipment.Provider.ShipmentProvider.Key 1421
Ektron.Cms.Commerce.Shipment.Provider.ShipmentProvider.MeterNumber 1421
Ektron.Cms.Commerce.Shipment.Provider.ShipmentProvider.Password 1421
Ektron.Cms.Commerce.Shipment.Provider.ShipmentProvider.ServiceUrl 1422
Ektron.Cms.Commerce.Shipment.Provider.ShipmentProvider.TransactionId 1422
Ektron.Cms.Commerce.ShippingBaseData 1559
Ektron.Cms.Commerce.ShippingBaseData.Id 1559
Ektron.Cms.Commerce.ShippingBaseData.Name 1560
Ektron.Cms.Commerce.ShippingBaseData.ProviderService 1560
Ektron.Cms.Commerce.ShippingBaseData.ShippingBaseData 1559
Ektron.Cms.Commerce.ShippingMethodApi 657
Ektron.Cms.Commerce.ShippingMethodApi.Add 658
Ektron.Cms.Commerce.ShippingMethodApi.Update 658
Ektron.Cms.Commerce.ShippingMethodData 1580
Ektron.Cms.Commerce.ShippingMethodData.DisplayOrder 1580
Ektron.Cms.Commerce.ShippingMethodData.IsActive 1581
Ektron.Cms.Commerce.ShippingMethodData.ShippingMethodData 1580
Ektron.Cms.Commerce.ShippingMethodProperty 1586
Ektron.Cms.Commerce.ShippingMethodProperty enumeration 1586
Ektron.Cms.Commerce.ShippingOptionData 1560
Ektron.Cms.Commerce.ShippingOptionData.CurrencyId 1561
Ektron.Cms.Commerce.ShippingOptionData.ShippingFee 1561
Ektron.Cms.Commerce.ShippingOptionData.ShippingOptionData 1560
Ektron.Cms.Commerce.TaxApi 658
Ektron.Cms.Commerce.TaxApi.Add 659
Ektron.Cms.Commerce.TaxApi.Delete 659
Ektron.Cms.Commerce.TaxApi.GetApplicableTaxRate 659
Ektron.Cms.Commerce.TaxApi.GetApplicableTaxRateList 660
Ektron.Cms.Commerce.TaxApi.GetItem 660
Ektron.Cms.Commerce.TaxApi.GetItemByCountryId 661
Ektron.Cms.Commerce.TaxApi.GetItemByPostalCode 661
Ektron.Cms.Commerce.TaxApi.GetItemByPostalId 662
Ektron.Cms.Commerce.TaxApi.GetItemByRegionId 662
Ektron.Cms.Commerce.TaxApi.GetList 662
Ektron.Cms.Commerce.TaxApi.Update 663
Ektron.Cms.Commerce.TaxClassApi 663
Ektron.Cms.Commerce.TaxClassApi.Add 663
Ektron.Cms.Commerce.TaxClassApi.Delete 664
Ektron.Cms.Commerce.TaxClassApi.GetItem 664
Ektron.Cms.Commerce.TaxClassApi.GetList 664
Ektron.Cms.Commerce.TaxClassApi.IsUsed 665
Ektron.Cms.Commerce.TaxClassApi.Update 665
Ektron.Cms.Commerce.TaxClassData 1561
Ektron.Cms.Commerce.TaxClassData.Id 1562
Ektron.Cms.Commerce.TaxClassData.Name 1563
Ektron.Cms.Commerce.TaxClassData.TaxClassData 1562
Ektron.Cms.Commerce.TaxClassProperty 1587
Ektron.Cms.Commerce.TaxClassProperty enumeration 1587
Ektron.Cms.Commerce.TaxRateData 1563
Ektron.Cms.Commerce.TaxRateData.Id 1564
Ektron.Cms.Commerce.TaxRateData.Rate 1565
Ektron.Cms.Commerce.TaxRateData.TaxClassId 1565
Ektron.Cms.Commerce.TaxRateData.TaxRateData 1564
Ektron.Cms.Commerce.TaxRateData.TypeId 1565
Ektron.Cms.Commerce.TaxRateData.TypeItemId 1565
Ektron.Cms.Commerce.TaxRateProperty 1587
Ektron.Cms.Commerce.TaxRateProperty enumeration 1587
Ektron.Cms.Commerce.TaxRateType 1587
Ektron.Cms.Commerce.TaxRateType enumeration 1587
Ektron.Cms.Commerce.TaxTypeApi 665
Ektron.Cms.Commerce.TaxTypeApi.Add 666

Ektron.Cms.Commerce.TaxTypeApi.Delete 666
Ektron.Cms.Commerce.TaxTypeApi.GetItem 666
Ektron.Cms.Commerce.TaxTypeApi.GetList 666
Ektron.Cms.Commerce.TaxTypeApi.Update 667
Ektron.Cms.Commerce.TaxTypeData 1565
Ektron.Cms.Commerce.TaxTypeData.Id 1567
Ektron.Cms.Commerce.TaxTypeData.Name 1567
Ektron.Cms.Commerce.TaxTypeData.Precedence 1567
Ektron.Cms.Commerce.TaxTypeData.TaxTypeData 1566
Ektron.Cms.Commerce.TaxTypeProperty 1588
Ektron.Cms.Commerce.TaxTypeProperty enumeration 1588
Ektron.Cms.Commerce.TierPriceData 1570
Ektron.Cms.Commerce.TierPriceData.Clone 1572
Ektron.Cms.Commerce.TierPriceData.CompareTo 1572
Ektron.Cms.Commerce.TierPriceData.DataState 1573
Ektron.Cms.Commerce.TierPriceData.Id 1573
Ektron.Cms.Commerce.TierPriceData.Quantity 1573
Ektron.Cms.Commerce.TierPriceData.RecurrenceId 1573
Ektron.Cms.Commerce.TierPriceData.SalePrice 1574
Ektron.Cms.Commerce.TierPriceData.TierPriceData 1571, 1572
Ektron.Cms.Commerce.TierPriceData.Validate 1572
Ektron.Cms.Commerce.WarehouseApi 667
Ektron.Cms.Commerce.WarehouseApi.Add 668
Ektron.Cms.Commerce.WarehouseApi.Delete 668
Ektron.Cms.Commerce.WarehouseApi.GetDefaultWarehouse 668
Ektron.Cms.Commerce.WarehouseApi.GetItem 668
Ektron.Cms.Commerce.WarehouseApi.GetList 668
Ektron.Cms.Commerce.WarehouseApi.MarkAsDefault 669
Ektron.Cms.Commerce.WarehouseApi.Update 669
Ektron.Cms.Commerce.WarehouseData 1575
Ektron.Cms.Commerce.WarehouseData.Address 1577
Ektron.Cms.Commerce.WarehouseData.Id 1577
Ektron.Cms.Commerce.WarehouseData.IsDefaultWarehouse 1577
Ektron.Cms.Commerce.WarehouseData.Name 1577
Ektron.Cms.Commerce.WarehouseData.WarehouseData 1576
Ektron.Cms.Commerce.WarehouseProperty 1588
Ektron.Cms.Commerce.WarehouseProperty enumeration 1588
Ektron.Cms.Commerce.Weight 1577
Ektron.Cms.Commerce.Weight.Amount 1579
Ektron.Cms.Commerce.Weight.Clone 1579
Ektron.Cms.Commerce.Weight.CompareTo 1579
Ektron.Cms.Commerce.Weight.ConvertUnits 1579
Ektron.Cms.Commerce.Weight.Units 1580
Ektron.Cms.Commerce.Weight.Weight 1578
Ektron.Cms.Commerce.WeightUnit 1588
Ektron.Cms.Commerce.WeightUnit enumeration 1588
Ektron.Cms.Common 1589
Ektron.Cms.Common namespace 1589
Classes 1589
Ektron.Cms.Common.CommerceSettings 1589
Ektron.Cms.Common.CommerceSettings.ComplianceMode 1590
Ektron.Cms.Common.CommerceSettings.CurrencyAlphaISO 1590
Ektron.Cms.Common.CommerceSettings.CurrencyCultureCode 1590
Ektron.Cms.Common.CommerceSettings.CurrencyId 1591
Ektron.Cms.Common.CommerceSettings.CurrencySymbol 1591
Ektron.Cms.Common.CommerceSettings.DefaultCurrencyId 1591
Ektron.Cms.Common.CommerceSettings.ISOCurrencySymbol 1591
Ektron.Cms.Common.CommerceSettings.OrderProcessingDisabled 1592
Ektron.Cms.Common.CommerceSettings.PasswordHistory 1592
Ektron.Cms.Common.CommerceSettings.TestMode 1592
Ektron.Cms.Common.UrlAliasAutoData 1592
Ektron.Cms.Common.UrlAliasAutoData.AliasName 1593
Ektron.Cms.Common.UrlAliasAutoData.AutoAliasType 1593
Ektron.Cms.Common.UrlAliasAutoData.AutoId 1594
Ektron.Cms.Common.UrlAliasAutoData.Example 1594
Ektron.Cms.Common.UrlAliasAutoData.ExcludedPath 1594
Ektron.Cms.Common.UrlAliasAutoData.FileExtension 1594
Ektron.Cms.Common.UrlAliasAutoData.FormattedSelectedPath 1595
Ektron.Cms.Common.UrlAliasAutoData.HostName 1595
Ektron.Cms.Common.UrlAliasAutoData.IsEnabled 1595
Ektron.Cms.Common.UrlAliasAutoData.LanguageId 1595
Ektron.Cms.Common.UrlAliasAutoData.PageNameType 1595

Ektron.Cms.Common.UrlAliasAutoData.ReplacementCharacter 1596
Ektron.Cms.Common.UrlAliasAutoData.SelectedPath 1596
Ektron.Cms.Common.UrlAliasAutoData.SiteId 1596
Ektron.Cms.Common.UrlAliasAutoData.SourceId 1596
Ektron.Cms.Common.UrlAliasAutoData.SourceName 1597
Ektron.Cms.Common.UrlAliasAutoData.Target 1597
Ektron.Cms.Common.UrlAliasInfo 1597
Ektron.Cms.Common.UrlAliasInfo.AliasId 1598
Ektron.Cms.Common.UrlAliasInfo.CurrentAliasExtension 1598
Ektron.Cms.Common.UrlAliasInfo.CurrentAliasName 1598
Ektron.Cms.Common.UrlAliasInfo.PreviousAliasExtension 1598
Ektron.Cms.Common.UrlAliasInfo.PreviousAliasName 1599
Ektron.Cms.Common.UrlAliasManualData 1599
Ektron.Cms.Common.UrlAliasManualData.AliasId 1600
Ektron.Cms.Common.UrlAliasManualData.AliasName 1600
Ektron.Cms.Common.UrlAliasManualData.AliasPageName 1600
Ektron.Cms.Common.UrlAliasManualData.ContentId 1600
Ektron.Cms.Common.UrlAliasManualData.ContentLanguage 1601
Ektron.Cms.Common.UrlAliasManualData.ContentTitle 1601
Ektron.Cms.Common.UrlAliasManualData.DisplayAlias 1601
Ektron.Cms.Common.UrlAliasManualData.FileExtension 1601
Ektron.Cms.Common.UrlAliasManualData.HostName 1602
Ektron.Cms.Common.UrlAliasManualData.IsDefault 1602
Ektron.Cms.Common.UrlAliasManualData.IsEnabled 1602
Ektron.Cms.Common.UrlAliasManualData.LibraryId 1602
Ektron.Cms.Common.UrlAliasManualData.QueryString 1603
Ektron.Cms.Common.UrlAliasManualData.QueryStringAction 1603
Ektron.Cms.Common.UrlAliasManualData.SiteId 1603
Ektron.Cms.Common.UrlAliasManualData.Target 1603
Ektron.Cms.Common.UrlAliasRegExData 1604
Ektron.Cms.Common.UrlAliasRegExData.Expression 1604
Ektron.Cms.Common.UrlAliasRegExData.ExpressionMap 1605
Ektron.Cms.Common.UrlAliasRegExData.ExpressionName 1605
Ektron.Cms.Common.UrlAliasRegExData.IsEnabled 1605
Ektron.Cms.Common.UrlAliasRegExData.Priority 1605
Ektron.Cms.Common.UrlAliasRegExData.RegExId 1606
Ektron.Cms.Common.UrlAliasRegExData.SiteId 1606
Ektron.Cms.Common.UrlAliasRegExData.TransformedUrl 1606
Ektron.Cms.ContentRatingLabels 1611
Ektron.Cms.ContentRatingLabels.AlreadyRatedMessage 1611
Ektron.Cms.ContentRatingLabels.BadLabel 1612
Ektron.Cms.ContentRatingLabels.CharsRemainingLabel 1612
Ektron.Cms.ContentRatingLabels.CookiesRequiredMessage 1612
Ektron.Cms.ContentRatingLabels.FeedbackHeader 1612
Ektron.Cms.ContentRatingLabels.GoodLabel 1613
Ektron.Cms.ContentRatingLabels.RatingLevelLabel 1613
Ektron.Cms.ContentRatingLabels.ThankYouMessage 1613
Ektron.Cms.ContentRatingLabels.TotalRatingsLabel 1613
Ektron.Cms.ContentRatingLabels.UserCommentsHeader 1613
Ektron.Cms.ContentRatingLabels.ValidationMessage 1614
Ektron.Cms.Controls 7
Ektron.Cms.Controls namespace 7
Classes 12
Structs, Records, Enums 548
Ektron.Cms.Controls.ActiveTopics 15
Ektron.Cms.Controls.ActiveTopics.BoardID 17
Ektron.Cms.Controls.ActiveTopics.CacheInterval 17
Ektron.Cms.Controls.ActiveTopics.DisplayMode 18
Ektron.Cms.Controls.ActiveTopics.DynamicForumParameter 18
Ektron.Cms.Controls.ActiveTopics.DynamicGroupParameter 18
Ektron.Cms.Controls.ActiveTopics.DynamicThreadParameter 19
Ektron.Cms.Controls.ActiveTopics.Fill 16
Ektron.Cms.Controls.ActiveTopics.MaxNumber 19
Ektron.Cms.Controls.ActiveTopics.ToString 16
Ektron.Cms.Controls.ActiveTopics.URLPath 19
Ektron.Cms.Controls.ActiveTopics.Wrap 16
Ektron.Cms.Controls.ActivityStream 20
Ektron.Cms.Controls.ActivityStream.CacheInterval 21
Ektron.Cms.Controls.ActivityStream.DefaultObjectID 22
Ektron.Cms.Controls.ActivityStream.DisplayXslt 22
Ektron.Cms.Controls.ActivityStream.DynamicObjectParameter 22
Ektron.Cms.Controls.ActivityStream.EnablePaging 23
Ektron.Cms.Controls.ActivityStream.Fill 20
Ektron.Cms.Controls.ActivityStream.LoadData 20
Ektron.Cms.Controls.ActivityStream.MaxResults 23

Ektron.Cms.Controls.ActivityStream.ObjectType 24
Ektron.Cms.Controls.ActivityStream.ProfileParamName 24
Ektron.Cms.Controls.ActivityStream.RaiseCallbackEvent 21
Ektron.Cms.Controls.ActivityStream.Stylesheet 24
Ektron.Cms.Controls.ActivityStream.TemplateUserProfile 25
Ektron.Cms.Controls.ActivityStream.XmlDoc 25
Ektron.Cms.Controls.AnalyticsTracker 25
Ektron.Cms.Controls.AnalyticsTracker.DefaultContentID 26
Ektron.Cms.Controls.AnalyticsTracker.DynamicParameter 27
Ektron.Cms.Controls.AnalyticsTracker.EnableAnalytics 27
Ektron.Cms.Controls.AnalyticsTracker.Wrap 26
Ektron.Cms.Controls.AssetControl 27
Ektron.Cms.Controls.AssetControl.DefaultAssetID 29
Ektron.Cms.Controls.AssetControl.DefaultFolderID 29
Ektron.Cms.Controls.AssetControl.DynamicParameter 30
Ektron.Cms.Controls.AssetControl.Fill 28
Ektron.Cms.Controls.AssetControl.IsImage 30
Ektron.Cms.Controls.AssetControl.OverrideExtension 30
Ektron.Cms.Controls.AssetControl.TaxonomyID 31
Ektron.Cms.Controls.AssetControl.ToString 28
Ektron.Cms.Controls.AssetControl.UploadType 31
Ektron.Cms.Controls.Blog 32
Ektron.Cms.Controls.Blog.ArchiveMode 33
Ektron.Cms.Controls.Blog.BlogID 34
Ektron.Cms.Controls.Blog.CacheInterval 34
Ektron.Cms.Controls.Blog.DateToStart 35
Ektron.Cms.Controls.Blog.DefaultUserID 35
Ektron.Cms.Controls.Blog.DynamicParameter 35
Ektron.Cms.Controls.Blog.DynamicUserParameter 36
Ektron.Cms.Controls.Blog.EditorHeight 36
Ektron.Cms.Controls.Blog.EditorWidth 36
Ektron.Cms.Controls.Blog.Fill 32
Ektron.Cms.Controls.Blog.JavascriptEditorHTMLMode 37
Ektron.Cms.Controls.Blog.MaxResults 37
Ektron.Cms.Controls.Blog.PostParameter 37
Ektron.Cms.Controls.Blog.RecentPosts 38
Ektron.Cms.Controls.Blog.ShowHeader 38
Ektron.Cms.Controls.Blog.ShowRSS 38
Ektron.Cms.Controls.Blog.ToString 32
Ektron.Cms.Controls.Blog.Wrap 33
Ektron.Cms.Controls.BlogArchive 39
Ektron.Cms.Controls.BlogArchive.ArchiveMode 40
Ektron.Cms.Controls.BlogArchive.BlogID 40
Ektron.Cms.Controls.BlogArchive.CacheInterval 41
Ektron.Cms.Controls.BlogArchive.DynamicParameter 41
Ektron.Cms.Controls.BlogArchive.Fill 39
Ektron.Cms.Controls.BlogArchive.Text 41
Ektron.Cms.Controls.BlogArchive.Wrap 39
Ektron.Cms.Controls.BlogArchive.WrapTag 42
Ektron.Cms.Controls.BlogCalendar 42
Ektron.Cms.Controls.BlogCalendar.BlogID 43
Ektron.Cms.Controls.BlogCalendar.CacheInterval 43
Ektron.Cms.Controls.BlogCalendar.Fill 42
Ektron.Cms.Controls.BlogCalendar.Text 44
Ektron.Cms.Controls.BlogCalendar.Wrap 42
Ektron.Cms.Controls.BlogCalendar.WrapTag 44
Ektron.Cms.Controls.BlogCategories 44
Ektron.Cms.Controls.BlogCategories.BlogID 45
Ektron.Cms.Controls.BlogCategories.CacheInterval 46
Ektron.Cms.Controls.BlogCategories.Fill 45
Ektron.Cms.Controls.BlogCategories.Text 46
Ektron.Cms.Controls.BlogCategories.Wrap 45
Ektron.Cms.Controls.BlogCategories.WrapTag 46
Ektron.Cms.Controls.BlogEntries 47
Ektron.Cms.Controls.BlogEntries.BlogID 49
Ektron.Cms.Controls.BlogEntries.BlogPostParameter 50
Ektron.Cms.Controls.BlogEntries.CacheInterval 50
Ektron.Cms.Controls.BlogEntries.DateToStart 50
Ektron.Cms.Controls.BlogEntries.DefaultUserID 51
Ektron.Cms.Controls.BlogEntries.DisplayXslt 51
Ektron.Cms.Controls.BlogEntries.DynamicParameter 51
Ektron.Cms.Controls.BlogEntries.DynamicUserParameter 52
Ektron.Cms.Controls.BlogEntries.EditorHeight 52
Ektron.Cms.Controls.BlogEntries.EditorWidth 52
Ektron.Cms.Controls.BlogEntries.Fill 47
Ektron.Cms.Controls.BlogEntries.GetRssFeed 47
Ektron.Cms.Controls.BlogEntries.GetView 48
Ektron.Cms.Controls.BlogEntries.GetViewNames 48

Ektron.Cms.Controls.BlogEntries.JavascriptEditorHTMLMode 53
Ektron.Cms.Controls.BlogEntries.MaxResults 53
Ektron.Cms.Controls.BlogEntries.ShowHeader 54
Ektron.Cms.Controls.BlogEntries.ToString 48
Ektron.Cms.Controls.BlogEntries.Wrap 48
Ektron.Cms.Controls.BlogEntries.XmlDoc 54
Ektron.Cms.Controls.BlogPost 54
Ektron.Cms.Controls.BlogPost.DefaultContentID 55
Ektron.Cms.Controls.BlogPost.ShowType 56
Ektron.Cms.Controls.BlogPost.Wrap 55
Ektron.Cms.Controls.BlogPost.XmlDoc 56
Ektron.Cms.Controls.BlogRecentPosts 56
Ektron.Cms.Controls.BlogRecentPosts.BlogID 57
Ektron.Cms.Controls.BlogRecentPosts.CacheInterval 58
Ektron.Cms.Controls.BlogRecentPosts.Fill 57
Ektron.Cms.Controls.BlogRecentPosts.NumberofPosts 58
Ektron.Cms.Controls.BlogRecentPosts.Text 59
Ektron.Cms.Controls.BlogRecentPosts.Wrap 57
Ektron.Cms.Controls.BlogRecentPosts.WrapTag 59
Ektron.Cms.Controls.BlogRoll 59
Ektron.Cms.Controls.BlogRoll.BlogID 60
Ektron.Cms.Controls.BlogRoll.CacheInterval 61
Ektron.Cms.Controls.BlogRoll.Fill 59
Ektron.Cms.Controls.BlogRoll.Text 61
Ektron.Cms.Controls.BlogRoll.Wrap 60
Ektron.Cms.Controls.BlogRoll.WrapTag 61
Ektron.Cms.Controls.BlogRSS 61
Ektron.Cms.Controls.BlogRSS.BlogID 62
Ektron.Cms.Controls.BlogRSS.CacheInterval 63
Ektron.Cms.Controls.BlogRSS.Fill 62
Ektron.Cms.Controls.BlogRSS.Text 63
Ektron.Cms.Controls.BlogRSS.WrapTag 63
Ektron.Cms.Controls.BreadCrumb 63
Ektron.Cms.Controls.BreadCrumb.CurrentPageIndicator 66
Ektron.Cms.Controls.BreadCrumb.DefaultContentID 67
Ektron.Cms.Controls.BreadCrumb.DisplayStyle 67
Ektron.Cms.Controls.BreadCrumb.DisplayTitle 67
Ektron.Cms.Controls.BreadCrumb.DynamicParameter 67
Ektron.Cms.Controls.BreadCrumb.Fill 64
Ektron.Cms.Controls.BreadCrumb.GenerateTrail_HTML 64
Ektron.Cms.Controls.BreadCrumb.IconAlt 68
Ektron.Cms.Controls.BreadCrumb.IconPath 68
Ektron.Cms.Controls.BreadCrumb.LinkLastItem 68
Ektron.Cms.Controls.BreadCrumb.LinkTarget 68
Ektron.Cms.Controls.BreadCrumb.MaxItems 69
Ektron.Cms.Controls.BreadCrumb.MetadataName 69
Ektron.Cms.Controls.BreadCrumb.Mode 69
Ektron.Cms.Controls.BreadCrumb.ResetTrail 65
Ektron.Cms.Controls.BreadCrumb.separator 70
Ektron.Cms.Controls.BreadCrumbItem 70
Ektron.Cms.Controls.BreadCrumbItem.BreadCrumbItem 70, 71
Ektron.Cms.Controls.BreadCrumbItem.CaseSensitiveUrlComparison
72
Ektron.Cms.Controls.BreadCrumbItem.DisplayTitle 72
Ektron.Cms.Controls.BreadCrumbItem.Equals 71
Ektron.Cms.Controls.BreadCrumbItem.GetHashCode 71
Ektron.Cms.Controls.BreadCrumbItem.IconAlt 72
Ektron.Cms.Controls.BreadCrumbItem.IconPath 73
Ektron.Cms.Controls.BreadCrumbItem.RawUrl 73
Ektron.Cms.Controls.BreadCrumbItem.Target 73
Ektron.Cms.Controls.BreadCrumbItem.ToString 72
Ektron.Cms.Controls.BusinessRules 73
Ektron.Cms.Controls.BusinessRules.Fill 74
Ektron.Cms.Controls.BusinessRules.Parameters 75
Ektron.Cms.Controls.BusinessRules.Result 75
Ektron.Cms.Controls.BusinessRules.RuleID 75
Ektron.Cms.Controls.BusinessRules.Text 76
Ektron.Cms.Controls.BusinessRules.Wrap 74
Ektron.Cms.Controls.BusinessRules.WrapTag 76
Ektron.Cms.Controls.Calendar 76
Ektron.Cms.Controls.Calendar.DefaultCalendarID 78
Ektron.Cms.Controls.Calendar.DisplayType 78
Ektron.Cms.Controls.Calendar.DynamicParameter 78
Ektron.Cms.Controls.Calendar.EndDate 79
Ektron.Cms.Controls.Calendar.Fill 77
Ektron.Cms.Controls.Calendar.HighlightEventID 79
Ektron.Cms.Controls.Calendar.HighlightEventTypeID 79
Ektron.Cms.Controls.Calendar.ShowXML 80

Ektron.Cms.Controls.Calendar.StartDate 80
Ektron.Cms.Controls.Calendar.Text 80
Ektron.Cms.Controls.Calendar.Wrap 77
Ektron.Cms.Controls.CalendarProvider 11
Ektron.Cms.Controls.CalendarProvider namespace 11
Ektron.Cms.Controls.Cart 81
Ektron.Cms.Controls.Cart.DisplayXslt 82
Ektron.Cms.Controls.Cart.DynamicProductParameter 82
Ektron.Cms.Controls.Cart.EnableCoupons 82
Ektron.Cms.Controls.Cart.LoadingImage 83
Ektron.Cms.Controls.Cart.Settings 83
Ektron.Cms.Controls.Cart.Stylesheet 83
Ektron.Cms.Controls.Cart.TemplateCheckout 84
Ektron.Cms.Controls.Cart.TemplateProduct 84
Ektron.Cms.Controls.Cart.TemplateShopping 85
Ektron.Cms.Controls.Cart.UserControlUrl 85
Ektron.Cms.Controls.Checkout 85
Ektron.Cms.Controls.Checkout.Cart 88
Ektron.Cms.Controls.Checkout.ControlPhase 86
Ektron.Cms.Controls.Checkout.ControlPhase enumeration 86
Ektron.Cms.Controls.Checkout.CurrentPhase 88
Ektron.Cms.Controls.Checkout.Customer 88
Ektron.Cms.Controls.Checkout.DefaultCountryId 89
Ektron.Cms.Controls.Checkout.DisplayXslt 89
Ektron.Cms.Controls.Checkout.DynamicPhaseParameter 89
Ektron.Cms.Controls.Checkout.Fill 86
Ektron.Cms.Controls.Checkout.FriendlyErrorMessage 90
Ektron.Cms.Controls.Checkout.IsSSLRequired 90
Ektron.Cms.Controls.Checkout.LoadingImage 91
Ektron.Cms.Controls.Checkout.PaymentMethod 91
Ektron.Cms.Controls.Checkout.PhaseChange 95
Ektron.Cms.Controls.Checkout.PhaseChangeEventHandler 94
Ektron.Cms.Controls.Checkout.PostLogin 95
Ektron.Cms.Controls.Checkout.PostLoginEventHandler 95
Ektron.Cms.Controls.Checkout.ShowErrorDetails 91
Ektron.Cms.Controls.Checkout.Stylesheet 92
Ektron.Cms.Controls.Checkout.TemplateCart 92
Ektron.Cms.Controls.Checkout.TemplateOrderHistory 92
Ektron.Cms.Controls.Checkout.TemplateRecoverPassword 93
Ektron.Cms.Controls.Checkout.TemplateShopping 93
Ektron.Cms.Controls.Checkout.ToString 86
Ektron.Cms.Controls.Checkout.UseMsAjax 94
Ektron.Cms.Controls.Checkout.ValidatePhase 94
Ektron.Cms.Controls.Checkout.Wrap 87
Ektron.Cms.Controls.Collection 95
Ektron.Cms.Controls.Collection.AddText 100
Ektron.Cms.Controls.Collection.CacheInterval 100
Ektron.Cms.Controls.Collection.CollectionDescription 100
Ektron.Cms.Controls.Collection.CollectionTitle 101
Ektron.Cms.Controls.Collection.ContentParameter 101
Ektron.Cms.Controls.Collection.CopyTo 96
Ektron.Cms.Controls.Collection.Count 101
Ektron.Cms.Controls.Collection.Current 101
Ektron.Cms.Controls.Collection.DefaultCollectionID 102
Ektron.Cms.Controls.Collection.DynamicParameter 102
Ektron.Cms.Controls.Collection.EkItems 102
Ektron.Cms.Controls.Collection.EnablePaging 103
Ektron.Cms.Controls.Collection.Fill 97
Ektron.Cms.Controls.Collection.GetEnumerator 97
Ektron.Cms.Controls.Collection.GetHtml 103
Ektron.Cms.Controls.Collection.GetRssFeed 97
Ektron.Cms.Controls.Collection.GetView 97
Ektron.Cms.Controls.Collection.GetViewNames 98
Ektron.Cms.Controls.Collection.IncludeIcons 103
Ektron.Cms.Controls.Collection.IsSynchronized 104
Ektron.Cms.Controls.Collection.LinkTarget 104
Ektron.Cms.Controls.Collection.MarkupLanguage 104
Ektron.Cms.Controls.Collection.MaxResults 104
Ektron.Cms.Controls.Collection.MoveNext 98
Ektron.Cms.Controls.Collection.Random 105
Ektron.Cms.Controls.Collection.Reset 98
Ektron.Cms.Controls.Collection.SelTaxonomyID 105
Ektron.Cms.Controls.Collection.SuppressWrapperTags 105
Ektron.Cms.Controls.Collection.SyncRoot 106
Ektron.Cms.Controls.Collection.this 98
Ektron.Cms.Controls.Collection.ToString 98
Ektron.Cms.Controls.Collection.Wrap 98
Ektron.Cms.Controls.Collection.XmlDoc 106

Ektron.Cms.Controls.Commerce 11

Ektron.Cms.Controls.Commerce namespace 11

Ektron.Cms.Controls.CommunityDocuments 106

Ektron.Cms.Controls.CommunityDocuments.AddItemFolderId 107

Ektron.Cms.Controls.CommunityDocuments.DefaultObjectID 108

Ektron.Cms.Controls.CommunityDocuments.DynamicContentBoxHeight 108

Ektron.Cms.Controls.CommunityDocuments.DynamicContentBoxWidth 108

Ektron.Cms.Controls.CommunityDocuments.DynamicContentTemplate 109

Ektron.Cms.Controls.CommunityDocuments.EnableAjax 109

Ektron.Cms.Controls.CommunityDocuments.EnableSearch 109

Ektron.Cms.Controls.CommunityDocuments.Fill 107

Ektron.Cms.Controls.CommunityDocuments.MemberMenuActive 109

Ektron.Cms.Controls.CommunityDocuments.ObjectType 110

Ektron.Cms.Controls.CommunityDocuments.Recursive 110

Ektron.Cms.Controls.CommunityDocuments.ShowAllChecked 110

Ektron.Cms.Controls.CommunityDocuments.Stylesheet 110

Ektron.Cms.Controls.CommunityDocuments.SuppressWrapperTags 111

Ektron.Cms.Controls.CommunityDocuments.TaxonomyDepth 111

Ektron.Cms.Controls.CommunityDocuments.TaxonomyId 111

Ektron.Cms.Controls.CommunityDocumentsObjectType 548

Ektron.Cms.Controls.CommunityDocumentsObjectType enumeration 548

Ektron.Cms.Controls.CommunityGroupBrowser 111

Ektron.Cms.Controls.CommunityGroupBrowser.AvatarHeight 113

Ektron.Cms.Controls.CommunityGroupBrowser.AvatarWidth 113

Ektron.Cms.Controls.CommunityGroupBrowser.CacheInterval 114

Ektron.Cms.Controls.CommunityGroupBrowser.DisplayXslt 114

Ektron.Cms.Controls.CommunityGroupBrowser.Fill 112

Ektron.Cms.Controls.CommunityGroupBrowser.Link 115

Ektron.Cms.Controls.CommunityGroupBrowser.LinkTarget 115

Ektron.Cms.Controls.CommunityGroupBrowser.MarkupLanguage 115

Ektron.Cms.Controls.CommunityGroupBrowser.MaxResults 115

Ektron.Cms.Controls.CommunityGroupBrowser.Stylesheet 116

Ektron.Cms.Controls.CommunityGroupBrowser.TagTemplate 116

Ektron.Cms.Controls.CommunityGroupBrowser.TagTemplateTarget 116

Ektron.Cms.Controls.CommunityGroupBrowser.TaxonomyId 116

Ektron.Cms.Controls.CommunityGroupBrowser.ToString 112

Ektron.Cms.Controls.CommunityGroupBrowser.Wrap 112

Ektron.Cms.Controls.CommunityGroupBrowser.XmlDoc 117

Ektron.Cms.Controls.CommunityGroupList 117

Ektron.Cms.Controls.CommunityGroupList.AllowGroupCreation 119

Ektron.Cms.Controls.CommunityGroupList.AvatarHeight 119

Ektron.Cms.Controls.CommunityGroupList.AvatarWidth 119

Ektron.Cms.Controls.CommunityGroupList.CacheInterval 119

Ektron.Cms.Controls.CommunityGroupList.DefaultUserID 120

Ektron.Cms.Controls.CommunityGroupList.DisplayMode 120

Ektron.Cms.Controls.CommunityGroupList.DisplayXslt 120

Ektron.Cms.Controls.CommunityGroupList.DynamicUserParameter 121

Ektron.Cms.Controls.CommunityGroupList.Fill 118

Ektron.Cms.Controls.CommunityGroupList.LanguageId 121

Ektron.Cms.Controls.CommunityGroupList.Link 122

Ektron.Cms.Controls.CommunityGroupList.LinkTarget 122

Ektron.Cms.Controls.CommunityGroupList.MarkupLanguage 122

Ektron.Cms.Controls.CommunityGroupList.MaxResults 122

Ektron.Cms.Controls.CommunityGroupList.ShowMySortingOptions 123

Ektron.Cms.Controls.CommunityGroupList.ShowSortingOptions 123

Ektron.Cms.Controls.CommunityGroupList.Stylesheet 123

Ektron.Cms.Controls.CommunityGroupList.TagTemplate 123

Ektron.Cms.Controls.CommunityGroupList.TagTemplateTarget 124

Ektron.Cms.Controls.CommunityGroupList.ToString 118
Ektron.Cms.Controls.CommunityGroupList.Wrap 118
Ektron.Cms.Controls.CommunityGroupList.XmlDoc 124
Ektron.Cms.Controls.CommunityGroupMembers 124
Ektron.Cms.Controls.CommunityGroupMembers.AvatarHeight 126
Ektron.Cms.Controls.CommunityGroupMembers.AvatarWidth 126
Ektron.Cms.Controls.CommunityGroupMembers.CacheInterval 126
Ektron.Cms.Controls.CommunityGroupMembers.DefaultCommunityGroupID 127
Ektron.Cms.Controls.CommunityGroupMembers.DisplayMode 127
Ektron.Cms.Controls.CommunityGroupMembers.DisplayXslt 127
Ektron.Cms.Controls.CommunityGroupMembers.DynamicParameter 128
Ektron.Cms.Controls.CommunityGroupMembers.Fill 125
Ektron.Cms.Controls.CommunityGroupMembers.Link 128
Ektron.Cms.Controls.CommunityGroupMembers.LinkTarget 128
Ektron.Cms.Controls.CommunityGroupMembers.MarkupLanguage 129
Ektron.Cms.Controls.CommunityGroupMembers.MaxResults 129
Ektron.Cms.Controls.CommunityGroupMembers.Stylesheet 129
Ektron.Cms.Controls.CommunityGroupMembers.ToString 125
Ektron.Cms.Controls.CommunityGroupMembers.Wrap 125
Ektron.Cms.Controls.CommunityGroupMembers.XmlDoc 130
Ektron.Cms.Controls.CommunityGroupProfile 130
Ektron.Cms.Controls.CommunityGroupProfile.DefaultGroupID 131
Ektron.Cms.Controls.CommunityGroupProfile.DisplayXslt 132
Ektron.Cms.Controls.CommunityGroupProfile.DynamicParameter 132
Ektron.Cms.Controls.CommunityGroupProfile.EkItem 132
Ektron.Cms.Controls.CommunityGroupProfile.Fill 130
Ektron.Cms.Controls.CommunityGroupProfile.GroupAdminLink 133
Ektron.Cms.Controls.CommunityGroupProfile.GroupAdminLinkTarget 133
Ektron.Cms.Controls.CommunityGroupProfile.Stylesheet 133
Ektron.Cms.Controls.CommunityGroupProfile.TagTemplate 133
Ektron.Cms.Controls.CommunityGroupProfile.TagTemplateTarget 134
Ektron.Cms.Controls.CommunityGroupProfile.Text 134
Ektron.Cms.Controls.CommunityGroupProfile.ToString 130
Ektron.Cms.Controls.CommunityGroupProfile.Wrap 131
Ektron.Cms.Controls.CommunityGroupProfile.XmlDoc 134
Ektron.Cms.Controls.CommunitySearch 135
Ektron.Cms.Controls.CommunitySearch.CacheInterval 137
Ektron.Cms.Controls.CommunitySearch.CustomPropertyID 138
Ektron.Cms.Controls.CommunitySearch.DisplayXslt 138
Ektron.Cms.Controls.CommunitySearch.EnableFilteringResults 139
Ektron.Cms.Controls.CommunitySearch.EnableGroupResults 139
Ektron.Cms.Controls.CommunitySearch.EnableMap 139
Ektron.Cms.Controls.CommunitySearch.EnableMouseOutHidesInfo 140
Ektron.Cms.Controls.CommunitySearch.EnableMouseOver 140
Ektron.Cms.Controls.CommunitySearch.EnableUnauthorizedSearching 140
Ektron.Cms.Controls.CommunitySearch.EnableUserResults 141
Ektron.Cms.Controls.CommunitySearch.Fill 135
Ektron.Cms.Controls.CommunitySearch.FriendsOnly 141
Ektron.Cms.Controls.CommunitySearch.GroupTaxonomyID 141
Ektron.Cms.Controls.CommunitySearch.InfoBoxShowsEmail 142
Ektron.Cms.Controls.CommunitySearch.LoadingImage 142
Ektron.Cms.Controls.CommunitySearch.MapControl 142
Ektron.Cms.Controls.CommunitySearch.MarkupLanguage 143
Ektron.Cms.Controls.CommunitySearch.MaxTagCount 143
Ektron.Cms.Controls.CommunitySearch.MembersOnly 143
Ektron.Cms.Controls.CommunitySearch.PageSize 144
Ektron.Cms.Controls.CommunitySearch.ResultProfile 144
Ektron.Cms.Controls.CommunitySearch.SearchPrivate 144
Ektron.Cms.Controls.CommunitySearch.ShowNameEmail 145
Ektron.Cms.Controls.CommunitySearch.StartingTab 145
Ektron.Cms.Controls.CommunitySearch.Stylesheet 146

Ektron.Cms.Controls.CommunitySearch.TemplateGroupProfile 146
Ektron.Cms.Controls.CommunitySearch.TemplateParamName 146
Ektron.Cms.Controls.CommunitySearch.TemplateTarget 147
Ektron.Cms.Controls.CommunitySearch.TemplateUserProfile 147
Ektron.Cms.Controls.CommunitySearch.ToString 135
Ektron.Cms.Controls.CommunitySearch.UseMsAjax 147
Ektron.Cms.Controls.CommunitySearch.UserTaxonomyID 148
Ektron.Cms.Controls.CommunitySearch.Wrap 136
Ektron.Cms.Controls.CommunitySearch.XmlDoc 148
Ektron.Cms.Controls.CommunitySearch.XsltDoc 148
Ektron.Cms.Controls.ContentBlock 148
Ektron.Cms.Controls.ContentBlock.CacheInterval 151
Ektron.Cms.Controls.ContentBlock.CopyTo 149
Ektron.Cms.Controls.ContentBlock.Count 152
Ektron.Cms.Controls.ContentBlock.Current 152
Ektron.Cms.Controls.ContentBlock.DefaultContentID 152
Ektron.Cms.Controls.ContentBlock.DisplayXslt 153
Ektron.Cms.Controls.ContentBlock.DynamicParameter 153
Ektron.Cms.Controls.ContentBlock.EkItem 153
Ektron.Cms.Controls.ContentBlock.Fill 150
Ektron.Cms.Controls.ContentBlock.GetEnumerator 150
Ektron.Cms.Controls.ContentBlock.IsSynchronized 154
Ektron.Cms.Controls.ContentBlock.MoveNext 150
Ektron.Cms.Controls.ContentBlock.OverrideXslt 154
Ektron.Cms.Controls.ContentBlock.Reset 150
Ektron.Cms.Controls.ContentBlock.StartingFolderID 154
Ektron.Cms.Controls.ContentBlock.SyncRoot 155
Ektron.Cms.Controls.ContentBlock.this 150
Ektron.Cms.Controls.ContentBlock.ToString 150
Ektron.Cms.Controls.ContentBlock.Wrap 151
Ektron.Cms.Controls.ContentFlagging 155
Ektron.Cms.Controls.ContentFlagging.DefaultContentID 156
Ektron.Cms.Controls.ContentFlagging.DisplayXslt 157
Ektron.Cms.Controls.ContentFlagging.DynamicParameter 157
Ektron.Cms.Controls.ContentFlagging.Fill 155
Ektron.Cms.Controls.ContentFlagging.FlagImage 157
Ektron.Cms.Controls.ContentFlagging.FlagText 158
Ektron.Cms.Controls.ContentFlagging.Stylesheet 158
Ektron.Cms.Controls.ContentFlagging.ToString 155
Ektron.Cms.Controls.ContentFlagging.Wrap 156
Ektron.Cms.Controls.ContentFlagging.XmlDoc 159
Ektron.Cms.Controls.ContentList 159
Ektron.Cms.Controls.ContentList.CacheInterval 163
Ektron.Cms.Controls.ContentList.ContentIds 163
Ektron.Cms.Controls.ContentList.CopyTo 160
Ektron.Cms.Controls.ContentList.Count 164
Ektron.Cms.Controls.ContentList.Current 164
Ektron.Cms.Controls.ContentList.DefaultContentID 164
Ektron.Cms.Controls.ContentList.Direction 164
Ektron.Cms.Controls.ContentList.DynamicParameter 165
Ektron.Cms.Controls.ContentList.EkItems 165
Ektron.Cms.Controls.ContentList.Fill 160
Ektron.Cms.Controls.ContentList.GetEnumerator 160
Ektron.Cms.Controls.ContentList.GetHtml 165
Ektron.Cms.Controls.ContentList.GetRssFeed 160
Ektron.Cms.Controls.ContentList.IncludeIcons 166
Ektron.Cms.Controls.ContentList.IsSynchronized 166
Ektron.Cms.Controls.ContentList.LinkTarget 166
Ektron.Cms.Controls.ContentList.MarkupLanguage 167
Ektron.Cms.Controls.ContentList.MetaTag 167
Ektron.Cms.Controls.ContentList.MoveNext 160
Ektron.Cms.Controls.ContentList.OrderByDirection 167
Ektron.Cms.Controls.ContentList.OrderKey 168
Ektron.Cms.Controls.ContentList.Random 168
Ektron.Cms.Controls.ContentList.Reset 161
Ektron.Cms.Controls.ContentList.SuppressWrapperTags 168
Ektron.Cms.Controls.ContentList.SyncRoot 169
Ektron.Cms.Controls.ContentList.this 161
Ektron.Cms.Controls.ContentList.ToString 161
Ektron.Cms.Controls.ContentList.Wrap 161
Ektron.Cms.Controls.ContentList.XmlDoc 169
Ektron.Cms.Controls.ContentRating 169
Ektron.Cms.Controls.ContentRating.AlreadyRatedMessage 171
Ektron.Cms.Controls.ContentRating.BadLabel 171
Ektron.Cms.Controls.ContentRating.CharsRemainingLabel 172
Ektron.Cms.Controls.ContentRating.CompleteGraphDisplay 172

Ektron.Cms.Controls.ContentRating.CookiesRequiredMessage 172
Ektron.Cms.Controls.ContentRating.DefaultContentID 173
Ektron.Cms.Controls.ContentRating.DynamicParameter 173
Ektron.Cms.Controls.ContentRating.FeedbackHeader 173
Ektron.Cms.Controls.ContentRating.Fill 170
Ektron.Cms.Controls.ContentRating.GoodLabel 174
Ektron.Cms.Controls.ContentRating.GraphBarColor 174
Ektron.Cms.Controls.ContentRating.InitialGraphDisplay 174
Ektron.Cms.Controls.ContentRating.Padding 175
Ektron.Cms.Controls.ContentRating.RatingLevelLabel 175
Ektron.Cms.Controls.ContentRating.ThankYouMessage 176
Ektron.Cms.Controls.ContentRating.ToString 170
Ektron.Cms.Controls.ContentRating.TotalRatingsLabel 176
Ektron.Cms.Controls.ContentRating.UserCommentsHeader 176
Ektron.Cms.Controls.ContentRating.ValidationMessage 177
Ektron.Cms.Controls.ContentRating.Wrap 170
Ektron.Cms.Controls.ContentReview 177
Ektron.Cms.Controls.ContentReview.DefaultContentID 179
Ektron.Cms.Controls.ContentReview.DisplayXslt 180
Ektron.Cms.Controls.ContentReview.DynamicParameter 180
Ektron.Cms.Controls.ContentReview.DynamicUserParameter 181
Ektron.Cms.Controls.ContentReview.Fill 178
Ektron.Cms.Controls.ContentReview.GetReviews 181
Ektron.Cms.Controls.ContentReview.MaxReviews 181
Ektron.Cms.Controls.ContentReview.Moderate 182
Ektron.Cms.Controls.ContentReview.RaiseCallbackEvent 178
Ektron.Cms.Controls.ContentReview.RatingsMinimum 182
Ektron.Cms.Controls.ContentReview.ToString 178
Ektron.Cms.Controls.ContentReview.UserID 182
Ektron.Cms.Controls.ContentReview.Wrap 178
Ektron.Cms.Controls.ContentReview.XmlDoc 183
Ektron.Cms.Controls.CurrencySelect 183
Ektron.Cms.Controls.CurrencySelect.CacheInterval 184
Ektron.Cms.Controls.CurrencySelect.DisplayXslt 185
Ektron.Cms.Controls.CurrencySelect.EkItem 186
Ektron.Cms.Controls.CurrencySelect.Fill 184
Ektron.Cms.Controls.CurrencySelect.Stylesheet 186
Ektron.Cms.Controls.CurrencySelect.ToString 184
Ektron.Cms.Controls.CurrencySelect.Wrap 184
Ektron.Cms.Controls.DesignTimeDiagnostic 186
Ektron.Cms.Controls.DesignTimeDiagnostic.Authenticated 187
Ektron.Cms.Controls.DesignTimeDiagnostic.Fill 187
Ektron.Cms.Controls.DesignTimeDiagnostic.Hide 187
Ektron.Cms.Controls.DesignTimeDiagnostic.Language 188
Ektron.Cms.Controls.DesignTimeDiagnostic.SuppressWrapperTags 188
Ektron.Cms.Controls.DesignTimeDiagnostic.WrapTag 188
Ektron.Cms.Controls.DhtmlMenu 188
Ektron.Cms.Controls.DhtmlMenu.AccessKey 190
Ektron.Cms.Controls.DhtmlMenu.Align 190
Ektron.Cms.Controls.DhtmlMenu.BackColor 191
Ektron.Cms.Controls.DhtmlMenu.BorderColor 191
Ektron.Cms.Controls.DhtmlMenu.BorderStyle 191
Ektron.Cms.Controls.DhtmlMenu.BorderWidth 191
Ektron.Cms.Controls.DhtmlMenu.CssClass 192
Ektron.Cms.Controls.DhtmlMenu.DisplayXslt 192
Ektron.Cms.Controls.DhtmlMenu.Enabled 192
Ektron.Cms.Controls.DhtmlMenu.EnableIE6FormsFix 192
Ektron.Cms.Controls.DhtmlMenu.Fill 189
Ektron.Cms.Controls.DhtmlMenu.Height 193
Ektron.Cms.Controls.DhtmlMenu.HorizontalDisplay 193
Ektron.Cms.Controls.DhtmlMenu.MenuHoverOverColor 193
Ektron.Cms.Controls.DhtmlMenu.MenuStyle 194
Ektron.Cms.Controls.DhtmlMenu.Stylesheet 194
Ektron.Cms.Controls.DhtmlMenu.SubMenuBackColor 194
Ektron.Cms.Controls.DhtmlMenu.SubMenuHoverOverColor 194
Ektron.Cms.Controls.DhtmlMenu.SubMenuItemHeight 195
Ektron.Cms.Controls.DhtmlMenu.SubMenuItemWidth 195
Ektron.Cms.Controls.DhtmlMenu.SubMenuTopOffset 195
Ektron.Cms.Controls.DhtmlMenu.TabIndex 196
Ektron.Cms.Controls.DhtmlMenu.ToolTip 196
Ektron.Cms.Controls.DhtmlMenu.Width 196
Ektron.Cms.Controls.DhtmlMenu.Wrap 189
Ektron.Cms.Controls.Directory 196
Ektron.Cms.Controls.Directory.AddItemFolderId 201
Ektron.Cms.Controls.Directory.CacheInterval 201
Ektron.Cms.Controls.Directory.CopyTo 197

Ektron.Cms.Controls.Directory.Count 202
Ektron.Cms.Controls.Directory.Current 202
Ektron.Cms.Controls.Directory.DisplayXslt 202
Ektron.Cms.Controls.Directory.DynamicParameter 202
Ektron.Cms.Controls.Directory.EkItems 203
Ektron.Cms.Controls.Directory.EnableAjax 203
Ektron.Cms.Controls.Directory.EnablePaging 203
Ektron.Cms.Controls.Directory.EnableSearch 204
Ektron.Cms.Controls.Directory.Fill 197
Ektron.Cms.Controls.Directory.GetEnumerator 198
Ektron.Cms.Controls.Directory.GetView 198
Ektron.Cms.Controls.Directory.GetViewNames 198
Ektron.Cms.Controls.Directory.IncludeIcons 204
Ektron.Cms.Controls.Directory.IsSynchronized 204
Ektron.Cms.Controls.Directory.ItemSortOrder 204
Ektron.Cms.Controls.Directory.LinkTarget 205
Ektron.Cms.Controls.Directory.MarkupLanguage 205
Ektron.Cms.Controls.Directory.MaxResults 205
Ektron.Cms.Controls.Directory.MoveNext 198
Ektron.Cms.Controls.Directory.Recursive 206
Ektron.Cms.Controls.Directory.Reset 199
Ektron.Cms.Controls.Directory.ShowAllChecked 206
Ektron.Cms.Controls.Directory.ShowCount 206
Ektron.Cms.Controls.Directory.ShowRoot 207
Ektron.Cms.Controls.Directory.SortDirection 207
Ektron.Cms.Controls.Directory.SuppressWrapperTags 207
Ektron.Cms.Controls.Directory.SyncRoot 208
Ektron.Cms.Controls.Directory.TaxonomyCols 208
Ektron.Cms.Controls.Directory.TaxonomyDepth 208
Ektron.Cms.Controls.Directory.TaxonomyDescription 208
Ektron.Cms.Controls.Directory.TaxonomyId 209
Ektron.Cms.Controls.Directory.TaxonomyItemCols 209
Ektron.Cms.Controls.Directory.TaxonomyParamId 209
Ektron.Cms.Controls.Directory.TaxonomyPath 209
Ektron.Cms.Controls.Directory.TaxonomyTitle 210
Ektron.Cms.Controls.Directory.TaxonomyTreeData 210
Ektron.Cms.Controls.Directory.TaxonomyXml 210
Ektron.Cms.Controls.Directory.this 199
Ektron.Cms.Controls.Directory.ToString 199
Ektron.Cms.Controls.Directory.Wrap 199
Ektron.Cms.Controls.Directory.XmlDoc 210
Ektron.Cms.Controls.ECommerce 11
Ektron.Cms.Controls.ECommerce namespace 11
Ektron.Cms.Controls.EktronCatalogPart 210
Ektron.Cms.Controls.EktronCatalogPart.BlogID 211
Ektron.Cms.Controls.EktronCatalogPart.BoardID 212
Ektron.Cms.Controls.EktronCatalogPart.CalendarDynamicParameter 212
Ektron.Cms.Controls.EktronCatalogPart.CollectionDynamicParameter 212
Ektron.Cms.Controls.EktronCatalogPart.ContentBlockDynamicParameter 213
Ektron.Cms.Controls.EktronCatalogPart.DefaultCalendarID 213
Ektron.Cms.Controls.EktronCatalogPart.DefaultCollectionID 213
Ektron.Cms.Controls.EktronCatalogPart.DefaultContentID 213
Ektron.Cms.Controls.EktronCatalogPart.DefaultFormID 214
Ektron.Cms.Controls.EktronCatalogPart.EnableBlogControl 214
Ektron.Cms.Controls.EktronCatalogPart.EnableCalendarControl 214
Ektron.Cms.Controls.EktronCatalogPart.EnableCollectionControl 215
Ektron.Cms.Controls.EktronCatalogPart.EnableContentBlockControl 215
Ektron.Cms.Controls.EktronCatalogPart.EnableDiscussionBoardsControl 215
Ektron.Cms.Controls.EktronCatalogPart.EnableFormBlockControl 215
Ektron.Cms.Controls.EktronCatalogPart.EnableListSummaryControl 216
Ektron.Cms.Controls.EktronCatalogPart.EnableThreadedDiscussionsControl 216
Ektron.Cms.Controls.EktronCatalogPart.FolderID 216
Ektron.Cms.Controls.EktronCatalogPart.FormBlockDynamicParameter 216
Ektron.Cms.Controls.EktronCatalogPart.Title 217
Ektron.Cms.Controls.EktronCommunityCatalogPart 217

Ektron.Cms.Controls.EktronCommunityCatalogPart.EnableBlogs 218
Ektron.Cms.Controls.EktronCommunityCatalogPart.EnableDiscussionBoards 218
Ektron.Cms.Controls.EktronCommunityCatalogPart.EnableForums 218
Ektron.Cms.Controls.EktronCommunityCatalogPart.Title 219
Ektron.Cms.Controls.EktronUserControl 12
Ektron.Cms.Controls.EktronUserControl namespace 12
Ektron.Cms.Controls.EktronWebPartZone 219
Ektron.Cms.Controls.EktronWebPartZone.CmsSitePath 219
Ektron.Cms.Controls.EkWebControl 220
Ektron.Cms.Controls.EkWebControl.Authenticated 225
Ektron.Cms.Controls.EkWebControl.CmsSitePath 226
Ektron.Cms.Controls.EkWebControl.EkWebControl 220
Ektron.Cms.Controls.EkWebControl.Error 226
Ektron.Cms.Controls.EkWebControl.Fill 221
Ektron.Cms.Controls.EkWebControl.GetEnumDescription 221
Ektron.Cms.Controls.EkWebControl.GetFilenameFromPath 221
Ektron.Cms.Controls.EkWebControl.GetMessage 222
Ektron.Cms.Controls.EkWebControl.Hide 226
Ektron.Cms.Controls.EkWebControl.IsLoggedIn 227
Ektron.Cms.Controls.EkWebControl.Language 227
Ektron.Cms.Controls.EkWebControl.LanguageID 227
Ektron.Cms.Controls.EkWebControl.LocalizeString 222
Ektron.Cms.Controls.EkWebControl.loggedInUserID 227
Ektron.Cms.Controls.EkWebControl.loggedInUserName 228
Ektron.Cms.Controls.EkWebControl.LogLevel 228
Ektron.Cms.Controls.EkWebControl.RedirectFromLoginPage 223
Ektron.Cms.Controls.EkWebControl.RedirectToLoginPage 223
Ektron.Cms.Controls.EkWebControl.RegisterAllDmsMenuCssAndScripts 223
Ektron.Cms.Controls.EkWebControl.RegisterDmsMenuCSS 223
Ektron.Cms.Controls.EkWebControl.RegisterDmsMenuScript 224
Ektron.Cms.Controls.EkWebControl.RegisterJQueryScript 224
Ektron.Cms.Controls.EkWebControl.RegisterMembershipTabScript 224
Ektron.Cms.Controls.EkWebControl.RegisterThickBoxCSS 224
Ektron.Cms.Controls.EkWebControl.RegisterThickBoxScript 224
Ektron.Cms.Controls.EkWebControl.SuppressWrapperTags 228
Ektron.Cms.Controls.EkWebControl.Text 228
Ektron.Cms.Controls.EkWebControl.ToString 224
Ektron.Cms.Controls.EkWebControl.Wrap 225
Ektron.Cms.Controls.EkWebControl.WrapTag 229
Ektron.Cms.Controls.EkWebPart 229
Ektron.Cms.Controls.EkWebPart.CatalogIconImageUrl 230
Ektron.Cms.Controls.EkWebPart.CreateEditorParts 230
Ektron.Cms.Controls.EkWebPart.Description 230
Ektron.Cms.Controls.EkWebPart.Subtitle 231
Ektron.Cms.Controls.EkWebPart.Title 231
Ektron.Cms.Controls.EkWebPart.TitleIconImageUrl 231
Ektron.Cms.Controls.EkWebPart.TitleUrl 231
Ektron.Cms.Controls.EkWebPart.WebBrowsableObject 231
Ektron.Cms.Controls.EkXsltBase 232
Ektron.Cms.Controls.EkXsltBase.CustomXml 232
Ektron.Cms.Controls.EkXsltBase.DisplayXslt 233
Ektron.Cms.Controls.EkXsltBase.EkXsltBase 232
Ektron.Cms.Controls.EkXsltBase.EnableErrorLogging 233
Ektron.Cms.Controls.EkXsltBase.XmlDoc 233
Ektron.Cms.Controls.EkXsltBase.XsltDoc 234
Ektron.Cms.Controls.ExplorerDragDrop 234
Ektron.Cms.Controls.ExplorerDragDrop.AssetID 234
Ektron.Cms.Controls.ExplorerDragDrop.ContentLanguage 235
Ektron.Cms.Controls.ExplorerDragDrop.FolderID 235
Ektron.Cms.Controls.ExplorerDragDrop.IsImage 235
Ektron.Cms.Controls.ExplorerDragDrop.OverrideExtension 236
Ektron.Cms.Controls.ExplorerDragDrop.TaxonomyId 236
Ektron.Cms.Controls.Favorites 236
Ektron.Cms.Controls.Favorites.CacheInterval 238
Ektron.Cms.Controls.Favorites.DefaultUserID 239
Ektron.Cms.Controls.Favorites.DisplayMode 239
Ektron.Cms.Controls.Favorites.DisplayXslt 239
Ektron.Cms.Controls.Favorites.DynamicParameter 240
Ektron.Cms.Controls.Favorites.FavoritesTitle 240
Ektron.Cms.Controls.Favorites.FavoritesUriTitle 240
Ektron.Cms.Controls.Favorites.Fill 237

Ektron.Cms.Controls.Favorites.Link 240
Ektron.Cms.Controls.Favorites.LinkTarget 241
Ektron.Cms.Controls.Favorites.MarkupLanguage 241
Ektron.Cms.Controls.Favorites.MaxResults 241
Ektron.Cms.Controls.Favorites.Stylesheet 242
Ektron.Cms.Controls.Favorites.ToString 237
Ektron.Cms.Controls.Favorites.Wrap 237
Ektron.Cms.Controls.Favorites.XmlDoc 242
Ektron.Cms.Controls.FlexMenu 242
Ektron.Cms.Controls.FlexMenu.AutoCollapseBranches 244
Ektron.Cms.Controls.FlexMenu.CacheInterval 244
Ektron.Cms.Controls.FlexMenu.CacheIntervalAuthenticated 245
Ektron.Cms.Controls.FlexMenu.DisplayXslt 245
Ektron.Cms.Controls.FlexMenu.EnableAjax 245
Ektron.Cms.Controls.FlexMenu.EnableMouseOverPopUp 246
Ektron.Cms.Controls.FlexMenu.EnableSmartOpen 246
Ektron.Cms.Controls.FlexMenu.EnableSubmenuLinkPathMatching 246
Ektron.Cms.Controls.FlexMenu.EnableSubmenuTemplateMatching 247
Ektron.Cms.Controls.FlexMenu.Fill 243
Ektron.Cms.Controls.FlexMenu.LogInfo 247
Ektron.Cms.Controls.FlexMenu.MasterControlId 247
Ektron.Cms.Controls.FlexMenu.MenuDepth 248
Ektron.Cms.Controls.FlexMenu.SetDirty 243
Ektron.Cms.Controls.FlexMenu.StartCollapsed 248
Ektron.Cms.Controls.FlexMenu.StartLevel 248
Ektron.Cms.Controls.FlexMenu.Stylesheet 249
Ektron.Cms.Controls.FlexMenu.UseAliasRealId 249
Ektron.Cms.Controls.FlexMenu.XmlDoc 249
Ektron.Cms.Controls.FlexMenu.XsltDoc 250
Ektron.Cms.Controls.FolderBreadcrumb 250
Ektron.Cms.Controls.FolderBreadcrumb.DefaultContentID 251
Ektron.Cms.Controls.FolderBreadcrumb.DefaultFolderID 251
Ektron.Cms.Controls.FolderBreadcrumb.DisplayStyle 252
Ektron.Cms.Controls.FolderBreadcrumb.DisplayXslt 252
Ektron.Cms.Controls.FolderBreadcrumb.DynamicParameter 253
Ektron.Cms.Controls.FolderBreadcrumb.Fill 250
Ektron.Cms.Controls.FolderBreadcrumb.LinkTarget 253
Ektron.Cms.Controls.FolderBreadcrumb.Mode 253
Ektron.Cms.Controls.FolderBreadcrumb.separator 254
Ektron.Cms.Controls.FolderBreadcrumb.XmlDoc 254
Ektron.Cms.Controls.FormBlock 254
Ektron.Cms.Controls.FormBlock.AddValidation 257
Ektron.Cms.Controls.FormBlock.CacheInterval 258
Ektron.Cms.Controls.FormBlock.CopyTo 255
Ektron.Cms.Controls.FormBlock.Count 258
Ektron.Cms.Controls.FormBlock.Current 258
Ektron.Cms.Controls.FormBlock.DefaultFormID 259
Ektron.Cms.Controls.FormBlock.DynamicParameter 259
Ektron.Cms.Controls.FormBlock.EkItem 259
Ektron.Cms.Controls.FormBlock.Fields 260
Ektron.Cms.Controls.FormBlock.Fill 255
Ektron.Cms.Controls.FormBlock.FormName 260
Ektron.Cms.Controls.FormBlock.GetEnumerator 255
Ektron.Cms.Controls.FormBlock.GetView 255
Ektron.Cms.Controls.FormBlock.GetViewNames 256
Ektron.Cms.Controls.FormBlock.IncludeTags 260
Ektron.Cms.Controls.FormBlock.IsSynchronized 261
Ektron.Cms.Controls.FormBlock.MoveNext 256
Ektron.Cms.Controls.FormBlock.Reset 256
Ektron.Cms.Controls.FormBlock.StartingFolderID 261
Ektron.Cms.Controls.FormBlock.SyncRoot 261
Ektron.Cms.Controls.FormBlock.this 256
Ektron.Cms.Controls.FormBlock.ToString 256
Ektron.Cms.Controls.FormBlock.Wrap 257
Ektron.Cms.Controls.FormFieldItem 261
Ektron.Cms.Controls.FormFieldItem.Basetype 262
Ektron.Cms.Controls.FormFieldItem.ContentType 262
Ektron.Cms.Controls.FormFieldItem.DatalistItems 263
Ektron.Cms.Controls.FormFieldItem.Datatype 263
Ektron.Cms.Controls.FormFieldItem.DisplayName 263
Ektron.Cms.Controls.FormFieldItem.FieldName 263
Ektron.Cms.Controls.FormFieldItem.Language 264
Ektron.Cms.Controls.FormFieldItem.Value 264
Ektron.Cms.Controls.Forum 264
Ektron.Cms.Controls.Forum.AllowAnonymousPost 268
Ektron.Cms.Controls.Forum.BoardID 268

Ektron.Cms.Controls.Forum.CacheInterval 269
Ektron.Cms.Controls.Forum.CustomOrderBy 269
Ektron.Cms.Controls.Forum.DefaultObjectID 269
Ektron.Cms.Controls.Forum.DynamicForumParameter 270
Ektron.Cms.Controls.Forum.DynamicGroupParameter 270
Ektron.Cms.Controls.Forum.DynamicObjectParameter 270
Ektron.Cms.Controls.Forum.DynamicParameter 271
Ektron.Cms.Controls.Forum.DynamicThreadParameter 271
Ektron.Cms.Controls.Forum.EditorCSS 271
Ektron.Cms.Controls.Forum.EditorToolbar 271
Ektron.Cms.Controls.Forum.EnableForumQuickSelect 272
Ektron.Cms.Controls.Forum.EnableThreadRating 272
Ektron.Cms.Controls.Forum.ExcludeList 273
Ektron.Cms.Controls.Forum.Fill 265
Ektron.Cms.Controls.Forum.FilterXslt 273
Ektron.Cms.Controls.Forum.GetCallbackResult 265
Ektron.Cms.Controls.Forum.HideUserProfile 273
Ektron.Cms.Controls.Forum.HighlightedUserRole 274
Ektron.Cms.Controls.Forum.JavascriptEditorHeight 274
Ektron.Cms.Controls.Forum.JavascriptEditorHTMLMode 275
Ektron.Cms.Controls.Forum.JavascriptEditorToolbar 275
Ektron.Cms.Controls.Forum.JavascriptEditorWidth 276
Ektron.Cms.Controls.Forum.ObjectType 276
Ektron.Cms.Controls.Forum.OrderBy 276
Ektron.Cms.Controls.Forum.OrderDirection 277
Ektron.Cms.Controls.Forum.ProfileLink 277
Ektron.Cms.Controls.Forum.ProfileLinkTarget 278
Ektron.Cms.Controls.Forum.RaiseCallbackEvent 265
Ektron.Cms.Controls.Forum.ShowCategories 278
Ektron.Cms.Controls.Forum.TaxCategories 278
Ektron.Cms.Controls.Forum.TaxonomyCols 278
Ektron.Cms.Controls.Forum.TaxonomyMarkupLanguage 279
Ektron.Cms.Controls.Forum.Theme 279
Ektron.Cms.Controls.Forum.ToString 266
Ektron.Cms.Controls.Forum.Wrap 266
Ektron.Cms.Controls.Friends 279
Ektron.Cms.Controls.Friends.AvatarHeight 281
Ektron.Cms.Controls.Friends.AvatarWidth 282
Ektron.Cms.Controls.Friends.CacheInterval 282
Ektron.Cms.Controls.Friends.DefaultUserID 282
Ektron.Cms.Controls.Friends.DisplayMode 283
Ektron.Cms.Controls.Friends.DisplayXslt 283
Ektron.Cms.Controls.Friends.DynamicParameter 283
Ektron.Cms.Controls.Friends.Fill 280
Ektron.Cms.Controls.Friends.FriendLabel 284
Ektron.Cms.Controls.Friends.FriendsLabel 284
Ektron.Cms.Controls.Friends.Link 284
Ektron.Cms.Controls.Friends.LinkTarget 285
Ektron.Cms.Controls.Friends.MarkupLanguage 285
Ektron.Cms.Controls.Friends.MaxResults 285
Ektron.Cms.Controls.Friends.ShowSortingOptions 285
Ektron.Cms.Controls.Friends.Stylesheet 286
Ektron.Cms.Controls.Friends.ToString 280
Ektron.Cms.Controls.Friends.Wrap 280
Ektron.Cms.Controls.Friends.XmlDoc 286
Ektron.Cms.Controls.HtmlEditDisplay 286
Ektron.Cms.Controls.HtmlEditDisplay.BaseJavaScript 287
Ektron.Cms.Controls.HtmlEditDisplay.BeginJavaScript 288
Ektron.Cms.Controls.HtmlEditDisplay.EndJavaScript 288
Ektron.Cms.Controls.HtmlEditDisplay.MakeFieldNameFromId 288
Ektron.Cms.Controls.HtmlEditDisplay.RenderCMSFields 288
Ektron.Cms.Controls.HtmlEditDisplay.SelectedJavaScript 289
Ektron.Cms.Controls.HtmlEditor 289
Ektron.Cms.Controls.HtmlEditor.Archived 293
Ektron.Cms.Controls.HtmlEditor.Authenticated 294
Ektron.Cms.Controls.HtmlEditor.CmsSitePath 294
Ektron.Cms.Controls.HtmlEditor.ContentAction 295
Ektron.Cms.Controls.HtmlEditor.ContentID 295
Ektron.Cms.Controls.HtmlEditor.ContentTitle 296
Ektron.Cms.Controls.HtmlEditor.ContentType 296
Ektron.Cms.Controls.HtmlEditor.EditorReadyCall 297
Ektron.Cms.Controls.HtmlEditor.EnableUpload 297
Ektron.Cms.Controls.HtmlEditor.ExecuteContentAction 291
Ektron.Cms.Controls.HtmlEditor.HtmlEditor 290
Ektron.Cms.Controls.HtmlEditor.IncludeContentInfo 298
Ektron.Cms.Controls.HtmlEditor.IsLoggedIn 298
Ektron.Cms.Controls.HtmlEditor.Language 298
Ektron.Cms.Controls.HtmlEditor.loggedInUserID 299

Ektron.Cms.Controls.HtmlEditor.loggedInUserName 299
Ektron.Cms.Controls.HtmlEditor.Path 300
Ektron.Cms.Controls.HtmlEditor.PreferredType 300
Ektron.Cms.Controls.HtmlEditor.SafeEncode 301
Ektron.Cms.Controls.HtmlEditor.ShowCMSToolbar 301
Ektron.Cms.Controls.HtmlEditor.ShowTitle 302
Ektron.Cms.Controls.HtmlEditor.StyleSheet 302
Ektron.Cms.Controls.HtmlEditor.Text 303
Ektron.Cms.Controls.HtmlEditor.ToolbarLevel 303
Ektron.Cms.Controls.HtmlEditor.ToolbarResetCall 304
Ektron.Cms.Controls.HtmlEditor.ToString 291
Ektron.Cms.Controls.HtmlEditor.TranslationFile 304
Ektron.Cms.Controls.HtmlEditor.WorkareaMode 291
Ektron.Cms.Controls.ImageControl 305
Ektron.Cms.Controls.ImageControl.DefaultImageID 306
Ektron.Cms.Controls.ImageControl.DynamicParameter 306
Ektron.Cms.Controls.ImageControl.Fill 305
Ektron.Cms.Controls.ImageControl.FolderID 307
Ektron.Cms.Controls.ImageControl.TaxonomyID 307
Ektron.Cms.Controls.ImageControl.Title 308
Ektron.Cms.Controls.ImageControl.Wrap 306
Ektron.Cms.Controls.IndexSearch 308
Ektron.Cms.Controls.IndexSearch.AddParm 310
Ektron.Cms.Controls.IndexSearch.ButtonImgSrc 314
Ektron.Cms.Controls.IndexSearch.ButtonText 315
Ektron.Cms.Controls.IndexSearch.ContentParameter 315
Ektron.Cms.Controls.IndexSearch.CopyTo 310
Ektron.Cms.Controls.IndexSearch.Count 315
Ektron.Cms.Controls.IndexSearch.Current 316
Ektron.Cms.Controls.IndexSearch.CustomOrderBy 316
Ektron.Cms.Controls.IndexSearch.EkItems 316
Ektron.Cms.Controls.IndexSearch.EmptyResultMsg 316
Ektron.Cms.Controls.IndexSearch.EncodeContentHtml 316
Ektron.Cms.Controls.IndexSearch.Fill 310
Ektron.Cms.Controls.IndexSearch.FolderId 317
Ektron.Cms.Controls.IndexSearch.FormName 317
Ektron.Cms.Controls.IndexSearch.GetEnumerator 311
Ektron.Cms.Controls.IndexSearch.GetSearchParmDataType 311
Ektron.Cms.Controls.IndexSearch.GetSearchParmSearchType 311
Ektron.Cms.Controls.IndexSearch.GetXPathValues 311
Ektron.Cms.Controls.IndexSearch.IndexSearch 308
Ektron.Cms.Controls.IndexSearch.IsSynchronized 317
Ektron.Cms.Controls.IndexSearch.Labelafter 318
Ektron.Cms.Controls.IndexSearch.Labelbefore 318
Ektron.Cms.Controls.IndexSearch.LabelBetween 318
Ektron.Cms.Controls.IndexSearch.LabelContains 319
Ektron.Cms.Controls.IndexSearch.LabeldateBetween 319
Ektron.Cms.Controls.IndexSearch.LabelEqual 319
Ektron.Cms.Controls.IndexSearch.LabelExactPhrase 320
Ektron.Cms.Controls.IndexSearch.LabelGreaterThan 320
Ektron.Cms.Controls.IndexSearch.Labelinputdate 320
Ektron.Cms.Controls.IndexSearch.Labelinputnumber 320
Ektron.Cms.Controls.IndexSearch.LabelLessThan 321
Ektron.Cms.Controls.IndexSearch.Labelon 321
Ektron.Cms.Controls.IndexSearch.LabelSelect 321
Ektron.Cms.Controls.IndexSearch.LinkTarget 322
Ektron.Cms.Controls.IndexSearch.MaxResults 322
Ektron.Cms.Controls.IndexSearch.MoveNext 312
Ektron.Cms.Controls.IndexSearch.OrderBy 322
Ektron.Cms.Controls.IndexSearch.OrderByDirection 323
Ektron.Cms.Controls.IndexSearch.Recursive 323
Ektron.Cms.Controls.IndexSearch.Reset 312
Ektron.Cms.Controls.IndexSearch.Search 312
Ektron.Cms.Controls.IndexSearch.SearchParmXML 323
Ektron.Cms.Controls.IndexSearch.SearchPram 309
Ektron.Cms.Controls.IndexSearch.SearchPram structure 309
Ektron.Cms.Controls.IndexSearch.ShowSearchBoxAlways 324
Ektron.Cms.Controls.IndexSearch.SyncRoot 324
Ektron.Cms.Controls.IndexSearch.this 313
Ektron.Cms.Controls.IndexSearch.Weighted 324
Ektron.Cms.Controls.IndexSearch.Wrap 313
Ektron.Cms.Controls.IndexSearch.XmlConfigId 324
Ektron.Cms.Controls.IndexSearch.XmlDoc 325
Ektron.Cms.Controls.Invite 325
Ektron.Cms.Controls.Invite.DisplayXslt 326
Ektron.Cms.Controls.Invite.Fill 325
Ektron.Cms.Controls.Invite.FriendMessageId 327

Ektron.Cms.Controls.Invite.InviteMessageId 327
Ektron.Cms.Controls.Invite.OptionalMessageText 327
Ektron.Cms.Controls.Invite.Stylesheet 328
Ektron.Cms.Controls.Invite.ToString 326
Ektron.Cms.Controls.Invite.Wrap 326
Ektron.Cms.Controls.Invite.XmlDoc 328
Ektron.Cms.Controls.LanguageAPI 328
Ektron.Cms.Controls.LanguageAPI.CurrentLanguage 330
Ektron.Cms.Controls.LanguageAPI.CurrentLanguageID 330
Ektron.Cms.Controls.LanguageAPI.DefaultLanguage 331
Ektron.Cms.Controls.LanguageAPI.DefaultLanguageID 331
Ektron.Cms.Controls.LanguageAPI.Fill 329
Ektron.Cms.Controls.LanguageAPI.GetLanguage 329
Ektron.Cms.Controls.LanguageAPI.GetLanguageID 329
Ektron.Cms.Controls.LanguageAPI.IsValid 329, 330
Ektron.Cms.Controls.LanguageAPI.LangaugeIdList 331
Ektron.Cms.Controls.LanguageAPI.LangaugeTitleList 331
Ektron.Cms.Controls.LanguageAPI.LanguageAPI 328
Ektron.Cms.Controls.LanguageAPI.MultiLanguageEnabled 331
Ektron.Cms.Controls.LanguageAPI.OverrideDefault 332
Ektron.Cms.Controls.LanguageAPI.SiteLanguage 332
Ektron.Cms.Controls.LanguageAPI.SiteLanguageID 332
Ektron.Cms.Controls.LanguageSelect 333
Ektron.Cms.Controls.LanguageSelect.Fill 333
Ektron.Cms.Controls.LanguageSelect.LabelName 334
Ektron.Cms.Controls.LanguageSelect.Text 334
Ektron.Cms.Controls.LanguageSelect.Wrap 333
Ektron.Cms.Controls.LanguageSelect.WrapTag 334
Ektron.Cms.Controls.ListSummary 335
Ektron.Cms.Controls.ListSummary.AddText 339
Ektron.Cms.Controls.ListSummary.CacheInterval 340
Ektron.Cms.Controls.ListSummary.ContentParameter 340
Ektron.Cms.Controls.ListSummary.ContentType 340
Ektron.Cms.Controls.ListSummary.CopyTo 335
Ektron.Cms.Controls.ListSummary.Count 341
Ektron.Cms.Controls.ListSummary.Current 341
Ektron.Cms.Controls.ListSummary.Direction 341
Ektron.Cms.Controls.ListSummary.EkItems 342
Ektron.Cms.Controls.ListSummary.EnablePaging 342
Ektron.Cms.Controls.ListSummary.Fill 336
Ektron.Cms.Controls.ListSummary.FolderDescription 342
Ektron.Cms.Controls.ListSummary.FolderID 343
Ektron.Cms.Controls.ListSummary.FolderName 343
Ektron.Cms.Controls.ListSummary.FolderPath 343
Ektron.Cms.Controls.ListSummary.GetEnumerator 336
Ektron.Cms.Controls.ListSummary.GetHtml 343
Ektron.Cms.Controls.ListSummary.GetRssFeed 336
Ektron.Cms.Controls.ListSummary.GetView 336
Ektron.Cms.Controls.ListSummary.GetViewNames 337
Ektron.Cms.Controls.ListSummary.IncludeIcons 344
Ektron.Cms.Controls.ListSummary.IsSynchronized 344
Ektron.Cms.Controls.ListSummary.LinkTarget 344
Ektron.Cms.Controls.ListSummary.MarkupLanguage 345
Ektron.Cms.Controls.ListSummary.MaxResults 345
Ektron.Cms.Controls.ListSummary.MoveNext 337
Ektron.Cms.Controls.ListSummary.OrderByDirection 345
Ektron.Cms.Controls.ListSummary.OrderKey 346
Ektron.Cms.Controls.ListSummary.Random 346
Ektron.Cms.Controls.ListSummary.Recursive 346
Ektron.Cms.Controls.ListSummary.Reset 337
Ektron.Cms.Controls.ListSummary.SelTaxonomyID 347
Ektron.Cms.Controls.ListSummary.SuppressWrapperTags 347
Ektron.Cms.Controls.ListSummary.SyncRoot 347
Ektron.Cms.Controls.ListSummary.this 337
Ektron.Cms.Controls.ListSummary.ToString 338
Ektron.Cms.Controls.ListSummary.Wrap 338
Ektron.Cms.Controls.ListSummary.XmlDoc 347
Ektron.Cms.Controls.Login 347
Ektron.Cms.Controls.Login.AutoAddType 348
Ektron.Cms.Controls.Login.AutoLogin 349
Ektron.Cms.Controls.Login.Fill 348
Ektron.Cms.Controls.Login.OnlyAllowMemberLogin 349
Ektron.Cms.Controls.Login.Password 349
Ektron.Cms.Controls.Login.PromptLogout 349
Ektron.Cms.Controls.Login.SuppressHelpButton 350
Ektron.Cms.Controls.Login.Text 350
Ektron.Cms.Controls.Login.UserName 350
Ektron.Cms.Controls.Map 350

Ektron.Cms.Controls.Map.Address 352
Ektron.Cms.Controls.Map.ContentId 353
Ektron.Cms.Controls.Map.CustomIcon 353
Ektron.Cms.Controls.Map.DateSearch 353
Ektron.Cms.Controls.Map.DisplayTextResult 353
Ektron.Cms.Controls.Map.DistanceUnit 354
Ektron.Cms.Controls.Map.EnableSearchResult 354
Ektron.Cms.Controls.Map.Fill 351
Ektron.Cms.Controls.Map.FolderId 354
Ektron.Cms.Controls.Map.ForceCategory 355
Ektron.Cms.Controls.Map.GeoControl 355
Ektron.Cms.Controls.Map.GetCallbackResult 351
Ektron.Cms.Controls.Map.Latitude 355
Ektron.Cms.Controls.Map.Longitude 356
Ektron.Cms.Controls.Map.MapCategory 356
Ektron.Cms.Controls.Map.MapProvider 356
Ektron.Cms.Controls.Map.MapStyle 356
Ektron.Cms.Controls.Map.MarkupLanguage 357
Ektron.Cms.Controls.Map.MinZoomLevel 357
Ektron.Cms.Controls.Map.PageSize 357
Ektron.Cms.Controls.Map.RaiseCallbackEvent 351
Ektron.Cms.Controls.Map.Recursive 357
Ektron.Cms.Controls.Map.SearchData 358
Ektron.Cms.Controls.Map.SelectedCategoryLogicalType 358
Ektron.Cms.Controls.Map.StartZoomLevel 358
Ektron.Cms.Controls.Map.Stylesheet 358
Ektron.Cms.Controls.Map.TemplateParamName 359
Ektron.Cms.Controls.Map.TemplateUserProfile 359
Ektron.Cms.Controls.Map.TypeControl 359
Ektron.Cms.Controls.Map.ZoomControl 359
Ektron.Cms.Controls.Membership 360
Ektron.Cms.Controls.Membership.AllowAvatarUpload 361
Ektron.Cms.Controls.Membership.AutoLogin 362
Ektron.Cms.Controls.Membership.BoardID 362
Ektron.Cms.Controls.Membership.DisplayMode 362
Ektron.Cms.Controls.Membership.DynamicParameter 363
Ektron.Cms.Controls.Membership.Fill 360
Ektron.Cms.Controls.Membership.JavascriptEditorHeight 363
Ektron.Cms.Controls.Membership.JavascriptEditorWidth 363
Ektron.Cms.Controls.Membership.RedirectFailedURL 364
Ektron.Cms.Controls.Membership.RedirectSuccessURL 364
Ektron.Cms.Controls.Membership.RegisterButtonImg 364
Ektron.Cms.Controls.Membership.RegisterButtonText 364
Ektron.Cms.Controls.Membership.ResetButtonImg 365
Ektron.Cms.Controls.Membership.ResetButtonText 365
Ektron.Cms.Controls.Membership.ResetPasswordMessageId 365
Ektron.Cms.Controls.Membership.SaveButtonImg 365
Ektron.Cms.Controls.Membership.SaveButtonText 366
Ektron.Cms.Controls.Membership.ShowExtended 366
Ektron.Cms.Controls.Membership.ShowTerms 366
Ektron.Cms.Controls.Membership.TaxonomyId 366
Ektron.Cms.Controls.Membership.Text 367
Ektron.Cms.Controls.Membership.UserExistsMessage 367
Ektron.Cms.Controls.Membership.UserID 367
Ektron.Cms.Controls.Membership.UserSuccessMessage 367
Ektron.Cms.Controls.Membership.UserUpdateSuccessMessage 368
Ektron.Cms.Controls.Menu 368
Ektron.Cms.Controls.Menu.CacheInterval 369
Ektron.Cms.Controls.Menu.DefaultMenuID 370
Ektron.Cms.Controls.Menu.DisplayXslt 370
Ektron.Cms.Controls.Menu.DynamicParameter 370
Ektron.Cms.Controls.Menu.Fill 368
Ektron.Cms.Controls.Menu.Stylesheet 371
Ektron.Cms.Controls.Menu.SuppressAddEdit 371
Ektron.Cms.Controls.Menu.Wrap 369
Ektron.Cms.Controls.MessageBoard 371
Ektron.Cms.Controls.MessageBoard.CacheInterval 374
Ektron.Cms.Controls.MessageBoard.CommentBoxLocation 374
Ektron.Cms.Controls.MessageBoard.CustomSpamMethod 372
Ektron.Cms.Controls.MessageBoard.DefaultObjectID 374
Ektron.Cms.Controls.MessageBoard.DisplayXslt 375
Ektron.Cms.Controls.MessageBoard.DynamicObjectParameter 375
Ektron.Cms.Controls.MessageBoard.EnablePaging 375
Ektron.Cms.Controls.MessageBoard.Fill 372
Ektron.Cms.Controls.MessageBoard.MarkupLanguage 376
Ektron.Cms.Controls.MessageBoard.MaxResults 376

Ektron.Cms.Controls.MessageBoard.Moderate 377
Ektron.Cms.Controls.MessageBoard.ObjectType 377
Ektron.Cms.Controls.MessageBoard.ProfileParamName 377
Ektron.Cms.Controls.MessageBoard.RaiseCallbackEvent 372
Ektron.Cms.Controls.MessageBoard.ShowMaxCharacter 378
Ektron.Cms.Controls.MessageBoard.SpamControlType 378
Ektron.Cms.Controls.MessageBoard.SpamTimeSpan 378
Ektron.Cms.Controls.MessageBoard.Stylesheet 379
Ektron.Cms.Controls.MessageBoard.TemplateUserProfile 379
Ektron.Cms.Controls.MessageBoard.ToString 372
Ektron.Cms.Controls.MessageBoard.Wrap 373
Ektron.Cms.Controls.MessageBoard.XmlDoc 379
Ektron.Cms.Controls.Messaging 380
Ektron.Cms.Controls.Messaging.DisplayXslt 381
Ektron.Cms.Controls.Messaging.EnablePreSearch 382
Ektron.Cms.Controls.Messaging.Fill 380
Ektron.Cms.Controls.Messaging.FriendsOnly 382
Ektron.Cms.Controls.Messaging.GroupParamName 382
Ektron.Cms.Controls.Messaging.JavascriptEditorHeight 383
Ektron.Cms.Controls.Messaging.JavascriptEditorHTMLMode 383
Ektron.Cms.Controls.Messaging.JavascriptEditorToolbar 383
Ektron.Cms.Controls.Messaging.JavascriptEditorWidth 384
Ektron.Cms.Controls.Messaging.ProfileParamName 384
Ektron.Cms.Controls.Messaging.ProfileUrl 385
Ektron.Cms.Controls.Messaging.RecipientParamName 385
Ektron.Cms.Controls.Messaging.ShowSearchOptions 385
Ektron.Cms.Controls.Messaging.Stylesheet 386
Ektron.Cms.Controls.Messaging.ToString 380
Ektron.Cms.Controls.Messaging.UserSearchDisplayXslt 386
Ektron.Cms.Controls.Messaging.Wrap 380
Ektron.Cms.Controls.Messaging.XmlDoc 386
Ektron.Cms.Controls.MetaData 386
Ektron.Cms.Controls.MetaData.CacheInterval 388
Ektron.Cms.Controls.MetaData.DefaultContentID 388
Ektron.Cms.Controls.MetaData.DefaultItemList 389
Ektron.Cms.Controls.MetaData.DynamicParameter 389
Ektron.Cms.Controls.MetaData.Fill 387
Ektron.Cms.Controls.MetaData.GenerateDublinCore 389
Ektron.Cms.Controls.MetaDataList 390
Ektron.Cms.Controls.MetaDataList.CacheInterval 393
Ektron.Cms.Controls.MetaDataList.ContentType 394
Ektron.Cms.Controls.MetaDataList.CopyTo 391
Ektron.Cms.Controls.MetaDataList.Count 394
Ektron.Cms.Controls.MetaDataList.Current 395
Ektron.Cms.Controls.MetaDataList.EkItems 395
Ektron.Cms.Controls.MetaDataList.EnablePaging 395
Ektron.Cms.Controls.MetaDataList.ExactPhrase 395
Ektron.Cms.Controls.MetaDataList.Fill 391
Ektron.Cms.Controls.MetaDataList.FolderID 396
Ektron.Cms.Controls.MetaDataList.GetEnumerator 391
Ektron.Cms.Controls.MetaDataList.GetHtml 396
Ektron.Cms.Controls.MetaDataList.IncludeIcons 397
Ektron.Cms.Controls.MetaDataList.IsSynchronized 397
Ektron.Cms.Controls.MetaDataList.KeyWordName 397
Ektron.Cms.Controls.MetaDataList.KeyWordValue 398
Ektron.Cms.Controls.MetaDataList.KeyWordValueMatchAll 398
Ektron.Cms.Controls.MetaDataList.KeyWordValueSeparator 398
Ektron.Cms.Controls.MetaDataList.LinkTarget 399
Ektron.Cms.Controls.MetaDataList.MarkupLanguage 399
Ektron.Cms.Controls.MetaDataList.MaxNumber 399
Ektron.Cms.Controls.MetaDataList.MoveNext 391
Ektron.Cms.Controls.MetaDataList.OrderBy 400
Ektron.Cms.Controls.MetaDataList.Recursive 400
Ektron.Cms.Controls.MetaDataList.Reset 391
Ektron.Cms.Controls.MetaDataList.SortOrder 401
Ektron.Cms.Controls.MetaDataList.SuppressWrapperTags 401
Ektron.Cms.Controls.MetaDataList.SyncRoot 401
Ektron.Cms.Controls.MetaDataList.this 391
Ektron.Cms.Controls.MetaDataList.Wrap 392
Ektron.Cms.Controls.MetaDataList.XmlDoc 401
Ektron.Cms.Controls.MicroMessaging 401
Ektron.Cms.Controls.MicroMessaging.CacheInterval 404
Ektron.Cms.Controls.MicroMessaging.CustomSpamMethod 402
Ektron.Cms.Controls.MicroMessaging.DefaultObjectID 404
Ektron.Cms.Controls.MicroMessaging.DisplayXslt 405
Ektron.Cms.Controls.MicroMessaging.DynamicObjectParameter 405

Ektron.Cms.Controls.MicroMessaging.EnablePaging 405
Ektron.Cms.Controls.MicroMessaging.Fill 402
Ektron.Cms.Controls.MicroMessaging.MaxResults 406
Ektron.Cms.Controls.MicroMessaging.MaxSearchResults 406
Ektron.Cms.Controls.MicroMessaging.Mode 407
Ektron.Cms.Controls.MicroMessaging.ProfileParamName 407
Ektron.Cms.Controls.MicroMessaging.RaiseCallbackEvent 402
Ektron.Cms.Controls.MicroMessaging.SpamControlType 407
Ektron.Cms.Controls.MicroMessaging.SpamTimeSpan 408
Ektron.Cms.Controls.MicroMessaging.Stylesheet 408
Ektron.Cms.Controls.MicroMessaging.TemplateMessage 408
Ektron.Cms.Controls.MicroMessaging.TemplateUserProfile 409
Ektron.Cms.Controls.MicroMessaging.ToString 403
Ektron.Cms.Controls.MicroMessaging.Wrap 403
Ektron.Cms.Controls.MicroMessaging.XmlDoc 409
Ektron.Cms.Controls.MyAccount 409
Ektron.Cms.Controls.MyAccount.CustomPropertyID 411
Ektron.Cms.Controls.MyAccount.DefaultCountryId 411
Ektron.Cms.Controls.MyAccount.DisplayXslt 412
Ektron.Cms.Controls.MyAccount.EkItem 412
Ektron.Cms.Controls.MyAccount.Fill 410
Ektron.Cms.Controls.MyAccount.LoadingImage 412
Ektron.Cms.Controls.MyAccount.RedirectUrl 413
Ektron.Cms.Controls.MyAccount.Stylesheet 413
Ektron.Cms.Controls.MyAccount.ToString 410
Ektron.Cms.Controls.MyAccount.UseMsAjax 413
Ektron.Cms.Controls.MyAccount.Wrap 410
Ektron.Cms.Controls.OrderList 414
Ektron.Cms.Controls.OrderList.CacheInterval 415
Ektron.Cms.Controls.OrderList.DefaultUserID 416
Ektron.Cms.Controls.OrderList.DisplayXslt 416
Ektron.Cms.Controls.OrderList.DynamicOrderParameter 417
Ektron.Cms.Controls.OrderList.DynamicProductParameter 417
Ektron.Cms.Controls.OrderList.EkItem 417
Ektron.Cms.Controls.OrderList.Fill 414
Ektron.Cms.Controls.OrderList.PageSize 418
Ektron.Cms.Controls.OrderList.Stylesheet 418
Ektron.Cms.Controls.OrderList.TemplateProduct 418
Ektron.Cms.Controls.OrderList.ToString 415
Ektron.Cms.Controls.OrderList.Wrap 415
Ektron.Cms.Controls.Personalization 11
Ektron.Cms.Controls.Personalization namespace 11
Classes 11
Ektron.Cms.Controls.Personalization.EkEditorPart 12
Ektron.Cms.Controls.PersonalizationManager 419
Ektron.Cms.Controls.PersonalizationManager.AddContentExitText 420
Ektron.Cms.Controls.PersonalizationManager.AddContentText 420
Ektron.Cms.Controls.PersonalizationManager.CurrentUserButtonText 421
Ektron.Cms.Controls.PersonalizationManager.Display 421
Ektron.Cms.Controls.PersonalizationManager.Fill 419
Ektron.Cms.Controls.PersonalizationManager.HeaderBackColor 421
Ektron.Cms.Controls.PersonalizationManager.HeaderText 422
Ektron.Cms.Controls.PersonalizationManager.PersonalizeExitText 422
Ektron.Cms.Controls.PersonalizationManager.PersonalizeScopeText 422
Ektron.Cms.Controls.PersonalizationManager.PersonalizeStartText 423
Ektron.Cms.Controls.PersonalizationManager.PublicUserButtonText 423
Ektron.Cms.Controls.PersonalizationManager.ResetLinkText 423
Ektron.Cms.Controls.PersonalizationManager.ResetLinkToolTip 424
Ektron.Cms.Controls.PersonalizationManager.UrlCompareMode 424
Ektron.Cms.Controls.PhotoGallery 425
Ektron.Cms.Controls.PhotoGallery.Fill 425
Ektron.Cms.Controls.Poll 425
Ektron.Cms.Controls.Poll.EnableAjax 427
Ektron.Cms.Controls.Poll.Fill 426
Ektron.Cms.Controls.Poll.PollID 427
Ektron.Cms.Controls.Poll.WrapPoll 426

Ektron.Cms.Controls.PostHistory 428
Ektron.Cms.Controls.PostHistory.BoardID 429
Ektron.Cms.Controls.PostHistory.CacheInterval 429
Ektron.Cms.Controls.PostHistory.Fill 428
Ektron.Cms.Controls.PostHistory.MaxNumber 430
Ektron.Cms.Controls.PostHistory.ToString 428
Ektron.Cms.Controls.PostHistory.URLPath 430
Ektron.Cms.Controls.PostHistory.UserID 430
Ektron.Cms.Controls.PostHistory.Wrap 429
Ektron.Cms.Controls.PostLoginEventArgs 431
Ektron.Cms.Controls.Product 431
Ektron.Cms.Controls.Product.CacheInterval 433
Ektron.Cms.Controls.Product.DefaultProductID 434
Ektron.Cms.Controls.Product.DisplayXslt 434
Ektron.Cms.Controls.Product.DynamicParameter 434
Ektron.Cms.Controls.Product.EkItems 435
Ektron.Cms.Controls.Product.Fill 432
Ektron.Cms.Controls.Product.ImageGallery 435
Ektron.Cms.Controls.Product.OverrideXslt 436
Ektron.Cms.Controls.Product.RedirectToBasketPage 432
Ektron.Cms.Controls.Product.ShowAddToCart 436
Ektron.Cms.Controls.Product.Stylesheet 436
Ektron.Cms.Controls.Product.TemplateCart 437
Ektron.Cms.Controls.Product.ToString 432
Ektron.Cms.Controls.Product.Wrap 432
Ektron.Cms.Controls.ProductList 437
Ektron.Cms.Controls.ProductList.DisplayXslt 439
Ektron.Cms.Controls.ProductList.DynamicParameter 440
Ektron.Cms.Controls.ProductList.EkItems 440
Ektron.Cms.Controls.ProductList.Fill 438
Ektron.Cms.Controls.ProductList.IdList 440
Ektron.Cms.Controls.ProductList.LoadingImage 441
Ektron.Cms.Controls.ProductList.PageSize 441
Ektron.Cms.Controls.ProductList.PageSpan 441
Ektron.Cms.Controls.ProductList.ProductCriteria 442
Ektron.Cms.Controls.ProductList.Recursive 442
Ektron.Cms.Controls.ProductList.SortMode 442
Ektron.Cms.Controls.ProductList.SourceId 443
Ektron.Cms.Controls.ProductList.SourceType 443
Ektron.Cms.Controls.ProductList.Stylesheet 444
Ektron.Cms.Controls.ProductList.TaxonomyDepth 444
Ektron.Cms.Controls.ProductList.TemplateProduct 445
Ektron.Cms.Controls.ProductList.ToString 438
Ektron.Cms.Controls.ProductList.UseMsAjax 445
Ektron.Cms.Controls.ProductList.Wrap 438
Ektron.Cms.Controls.ProductSearch 445
Ektron.Cms.Controls.ProductSearch.ButtonImgSrc 449
Ektron.Cms.Controls.ProductSearch.ButtonText 450
Ektron.Cms.Controls.ProductSearch.CacheInterval 450
Ektron.Cms.Controls.ProductSearch.CatalogId 450
Ektron.Cms.Controls.ProductSearch.CustomOrderBy 451
Ektron.Cms.Controls.ProductSearch.DisplayXslt 451
Ektron.Cms.Controls.ProductSearch.DynamicProductParameter 451
Ektron.Cms.Controls.ProductSearch.EnableAdvancedLink 452
Ektron.Cms.Controls.ProductSearch.EndCount 452
Ektron.Cms.Controls.ProductSearch.Fill 446
Ektron.Cms.Controls.ProductSearch.GetCallbackResult 446
Ektron.Cms.Controls.ProductSearch.IsInWorkArea 452
Ektron.Cms.Controls.ProductSearch.LinkTarget 453
Ektron.Cms.Controls.ProductSearch.LoadingImage 453
Ektron.Cms.Controls.ProductSearch.MaxCharacters 453
Ektron.Cms.Controls.ProductSearch.MaxTeaserLength 454
Ektron.Cms.Controls.ProductSearch.OrderBy 454
Ektron.Cms.Controls.ProductSearch.OrderDirection 455
Ektron.Cms.Controls.ProductSearch.ProductSearch 446
Ektron.Cms.Controls.ProductSearch.RaiseCallbackEvent 447
Ektron.Cms.Controls.ProductSearch.Recursive 455
Ektron.Cms.Controls.ProductSearch.ResultCount 455
Ektron.Cms.Controls.ProductSearch.ResultDataTable 456
Ektron.Cms.Controls.ProductSearch.ResultsPageSize 456
Ektron.Cms.Controls.ProductSearch.ResultTagId 456
Ektron.Cms.Controls.ProductSearch.ResultXml 457
Ektron.Cms.Controls.ProductSearch.SearchText 457
Ektron.Cms.Controls.ProductSearch.ShowCategories 457
Ektron.Cms.Controls.ProductSearch.ShowCustomSummary 458
Ektron.Cms.Controls.ProductSearch.ShowSearchBoxAlways 458

Ektron.Cms.Controls.ProductSearch.StartCount 459
Ektron.Cms.Controls.ProductSearch.Stylesheet 459
Ektron.Cms.Controls.ProductSearch.TaxCategories 459
Ektron.Cms.Controls.ProductSearch.TaxonomyOperator 459
Ektron.Cms.Controls.ProductSearch.TemplateCart 460
Ektron.Cms.Controls.ProductSearch.TemplateProduct 460
Ektron.Cms.Controls.ProductSearch.TextBoxSize 461
Ektron.Cms.Controls.ProductSearch.TimeTaken 461
Ektron.Cms.Controls.ProductSearch.Wrap 447
Ektron.Cms.Controls.ProductSearch.XmlDoc 461
Ektron.Cms.Controls.Recommendation 462
Ektron.Cms.Controls.Recommendation.CacheInterval 463
Ektron.Cms.Controls.Recommendation.DefaultProductID 464
Ektron.Cms.Controls.Recommendation.DisplayXslt 464
Ektron.Cms.Controls.Recommendation.DynamicCartParameter 465
Ektron.Cms.Controls.Recommendation.DynamicProductParameter 465
Ektron.Cms.Controls.Recommendation.EkItem 466
Ektron.Cms.Controls.Recommendation.Fill 462
Ektron.Cms.Controls.Recommendation.PageSize 466
Ektron.Cms.Controls.Recommendation.RecommendationType 466
Ektron.Cms.Controls.Recommendation.Stylesheet 467
Ektron.Cms.Controls.Recommendation.TemplateCart 467
Ektron.Cms.Controls.Recommendation.TemplateProduct 467
Ektron.Cms.Controls.Recommendation.ToString 462
Ektron.Cms.Controls.Recommendation.Wrap 462
Ektron.Cms.Controls.RssAggregator 468
Ektron.Cms.Controls.RssAggregator.CopyTo 469
Ektron.Cms.Controls.RssAggregator.Count 471
Ektron.Cms.Controls.RssAggregator.Current 471
Ektron.Cms.Controls.RssAggregator.EkItems 471
Ektron.Cms.Controls.RssAggregator.Fill 469
Ektron.Cms.Controls.RssAggregator.GetEnumerator 469
Ektron.Cms.Controls.RssAggregator.GetView 469
Ektron.Cms.Controls.RssAggregator.GetViewNames 469
Ektron.Cms.Controls.RssAggregator.IsSynchronized 471
Ektron.Cms.Controls.RssAggregator.LinkTarget 472
Ektron.Cms.Controls.RssAggregator.MaxResults 472
Ektron.Cms.Controls.RssAggregator.MoveNext 470
Ektron.Cms.Controls.RssAggregator.Reset 470
Ektron.Cms.Controls.RssAggregator.SyncRoot 472
Ektron.Cms.Controls.RssAggregator.this 470
Ektron.Cms.Controls.RssAggregator.URL 473
Ektron.Cms.Controls.RssAggregator.WrapTag 473
Ektron.Cms.Controls.RssItemBase 473
Ektron.Cms.Controls.RssItemBase.DatePosted 474
Ektron.Cms.Controls.RssItemBase.Description 474
Ektron.Cms.Controls.RssItemBase.Link 474
Ektron.Cms.Controls.RssItemBase.Title 474
Ektron.Cms.Controls.Search 474
Ektron.Cms.Controls.Search.ButtonImgSrc 480
Ektron.Cms.Controls.Search.ButtonText 480
Ektron.Cms.Controls.Search.ContentFieldsetLegend 480
Ektron.Cms.Controls.Search.CopyTo 476
Ektron.Cms.Controls.Search.Count 481
Ektron.Cms.Controls.Search.Current 481
Ektron.Cms.Controls.Search.Display 481
Ektron.Cms.Controls.Search.EkItems 481
Ektron.Cms.Controls.Search.EmptyResultMsg 482
Ektron.Cms.Controls.Search.EnableAdvancedLink 482
Ektron.Cms.Controls.Search.EnableAjax 482
Ektron.Cms.Controls.Search.EnableArchived 483
Ektron.Cms.Controls.Search.EnableArchivedBtn 483
Ektron.Cms.Controls.Search.EnableContent 483
Ektron.Cms.Controls.Search.EnableContentBtn 483
Ektron.Cms.Controls.Search.EnableContentFieldset 484
Ektron.Cms.Controls.Search.EnableDMSAssets 484
Ektron.Cms.Controls.Search.EnableDMSAssetsBtn 484
Ektron.Cms.Controls.Search.EnableFolderFieldset 485
Ektron.Cms.Controls.Search.EnableForms 485
Ektron.Cms.Controls.Search.EnableFormsBtn 485
Ektron.Cms.Controls.Search.EnableLibFiles 486
Ektron.Cms.Controls.Search.EnableLibFilesBtn 486
Ektron.Cms.Controls.Search.EnableLibHyperlinks 486
Ektron.Cms.Controls.Search.EnableLibHyperlinksBtn 486
Ektron.Cms.Controls.Search.EnableLibImages 487
Ektron.Cms.Controls.Search.EnableLibImagesBtn 487

Ektron.Cms.Controls.Search.EnableLibraryFieldset 487
Ektron.Cms.Controls.Search.Fill 476
Ektron.Cms.Controls.Search.FolderFieldsetLegend 488
Ektron.Cms.Controls.Search.FolderID 488
Ektron.Cms.Controls.Search.FolderPath 488
Ektron.Cms.Controls.Search.GetEnumerator 476
Ektron.Cms.Controls.Search.IncludeIcons 489
Ektron.Cms.Controls.Search.IsSynchronized 489
Ektron.Cms.Controls.Search.LibraryFieldsetLegend 489
Ektron.Cms.Controls.Search.LinkTarget 490
Ektron.Cms.Controls.Search.LoadSearch 476
Ektron.Cms.Controls.Search.MaxCharacters 490
Ektron.Cms.Controls.Search.MaxResults 490
Ektron.Cms.Controls.Search.MaxTeaserLength 491
Ektron.Cms.Controls.Search.MoveNext 476
Ektron.Cms.Controls.Search.OrderBy 491
Ektron.Cms.Controls.Search.OrderDirection 491
Ektron.Cms.Controls.Search.PostResultsAt 492
Ektron.Cms.Controls.Search.Recursive 492
Ektron.Cms.Controls.Search.RemoveTeaserHtml 492
Ektron.Cms.Controls.Search.Reset 477
Ektron.Cms.Controls.Search.ResultTagId 493
Ektron.Cms.Controls.Search.Search 475
Ektron.Cms.Controls.Search.SearchDisplayUI 477
Ektron.Cms.Controls.Search.SearchText 493
Ektron.Cms.Controls.Search.SearchType 493
Ektron.Cms.Controls.Search.ShowDate 493
Ektron.Cms.Controls.Search.ShowExtendedSearch 494
Ektron.Cms.Controls.Search.ShowSearchBoxAlways 494
Ektron.Cms.Controls.Search.ShowSearchOptions 494
Ektron.Cms.Controls.Search.Stylesheet 495
Ektron.Cms.Controls.Search.SyncRoot 495
Ektron.Cms.Controls.Search.TargetFolderDescriptionList 495
Ektron.Cms.Controls.Search.TextBoxSize 496
Ektron.Cms.Controls.Search.this 477
Ektron.Cms.Controls.Search.Wrap 477
Ektron.Cms.Controls.Search.XmlDoc 496
Ektron.Cms.Controls.Sitemap 496
Ektron.Cms.Controls.Sitemap.CacheInterval 497
Ektron.Cms.Controls.Sitemap.ClassName 498
Ektron.Cms.Controls.Sitemap.DisplayType 498
Ektron.Cms.Controls.Sitemap.DisplayXslt 498
Ektron.Cms.Controls.Sitemap.Fill 496
Ektron.Cms.Controls.Sitemap.FlatTopLevel 499
Ektron.Cms.Controls.Sitemap.FolderID 499
Ektron.Cms.Controls.Sitemap.MaxLevel 499
Ektron.Cms.Controls.Sitemap.StartingLevel 500
Ektron.Cms.Controls.Sitemap.Stylesheet 500
Ektron.Cms.Controls.Sitemap.XmlDoc 500
Ektron.Cms.Controls.SmartMenu 500
Ektron.Cms.Controls.SmartMenu.AutoCollapseBranches 501
Ektron.Cms.Controls.SmartMenu.DisplayXslt 502
Ektron.Cms.Controls.SmartMenu.Enable508Compliance 502
Ektron.Cms.Controls.SmartMenu.Enable508NavigationHeadings 502
Ektron.Cms.Controls.SmartMenu.EnableMouseOverPopUp 503
Ektron.Cms.Controls.SmartMenu.EnableSmartOpen 503
Ektron.Cms.Controls.SmartMenu.EnableSubmenuLinkPathMatching 504
Ektron.Cms.Controls.SmartMenu.EnableSubmenuTemplateMatching 504
Ektron.Cms.Controls.SmartMenu.LaunchLinksInNewWindow 504
Ektron.Cms.Controls.SmartMenu.LogInfo 505
Ektron.Cms.Controls.SmartMenu.ShowRootFolder 505
Ektron.Cms.Controls.SmartMenu.SingleEditButton 505
Ektron.Cms.Controls.SmartMenu.StartCollapsed 506
Ektron.Cms.Controls.SmartMenu.StartWithRootCollapsed 506
Ektron.Cms.Controls.SmartMenu.Stylesheet 506
Ektron.Cms.Controls.SocialBar 507
Ektron.Cms.Controls.SocialBar.CacheInterval 509
Ektron.Cms.Controls.SocialBar.DefaultObjectID 509
Ektron.Cms.Controls.SocialBar.DisplayXslt 510
Ektron.Cms.Controls.SocialBar.DynamicObjectParameter 510
Ektron.Cms.Controls.SocialBar.EkItems 510
Ektron.Cms.Controls.SocialBar.Fill 507
Ektron.Cms.Controls.SocialBar.GetCallbackResult 508
Ektron.Cms.Controls.SocialBar.IncludeIcons 511

Ektron.Cms.Controls.SocialBar.InviteURL 511
Ektron.Cms.Controls.SocialBar.Items 511
Ektron.Cms.Controls.SocialBar.MarkupLanguage 512
Ektron.Cms.Controls.SocialBar.MessagingURL 512
Ektron.Cms.Controls.SocialBar.ObjectType 512
Ektron.Cms.Controls.SocialBar.Stylesheet 512
Ektron.Cms.Controls.SocialBar.SuppressWrapperTags 513
Ektron.Cms.Controls.SocialBar.ToString 508
Ektron.Cms.Controls.SocialBar.Wrap 508
Ektron.Cms.Controls.SocialBar.XmlDoc 513
Ektron.Cms.Controls.Support 12
Ektron.Cms.Controls.Support namespace 12
Ektron.Cms.Controls.TagCloud 513
Ektron.Cms.Controls.TagCloud.CacheInterval 515
Ektron.Cms.Controls.TagCloud.DisplayXslt 516
Ektron.Cms.Controls.TagCloud.DynamicParameter 516
Ektron.Cms.Controls.TagCloud.Fill 514
Ektron.Cms.Controls.TagCloud.MarkupLanguage 516
Ektron.Cms.Controls.TagCloud.MaxTagsDisplayed 517
Ektron.Cms.Controls.TagCloud.OrderBy 517
Ektron.Cms.Controls.TagCloud.OrderByDirection 517
Ektron.Cms.Controls.TagCloud.Stylesheet 518
Ektron.Cms.Controls.TagCloud.TagTemplate 518
Ektron.Cms.Controls.TagCloud.TagTemplateTarget 519
Ektron.Cms.Controls.TagCloud.TagType 519
Ektron.Cms.Controls.TagCloud.ToString 514
Ektron.Cms.Controls.TagCloud.Wrap 514
Ektron.Cms.Controls.TagCloud.XmlDoc 519
Ektron.Cms.Controls.UserProfile 520
Ektron.Cms.Controls.UserProfile.AvatarHeight 521
Ektron.Cms.Controls.UserProfile.AvatarWidth 521
Ektron.Cms.Controls.UserProfile.CacheInterval 522
Ektron.Cms.Controls.UserProfile.DefaultUserID 522
Ektron.Cms.Controls.UserProfile.DisplayXslt 523
Ektron.Cms.Controls.UserProfile.DynamicParameter 523
Ektron.Cms.Controls.UserProfile.EnableEmailDisplay 523
Ektron.Cms.Controls.UserProfile.Fill 520
Ektron.Cms.Controls.UserProfile.Stylesheet 524
Ektron.Cms.Controls.UserProfile.TagTemplate 524
Ektron.Cms.Controls.UserProfile.TagTemplateTarget 524
Ektron.Cms.Controls.UserProfile.TaxonomyId 524
Ektron.Cms.Controls.UserProfile.ToString 520
Ektron.Cms.Controls.UserProfile.Wrap 520
Ektron.Cms.Controls.UserProfile.XmlDoc 525
Ektron.Cms.Controls.WebCalendar 525
Ektron.Cms.Controls.WebCalendar.AllowEventEditing 526
Ektron.Cms.Controls.WebCalendar.CssFile 527
Ektron.Cms.Controls.WebCalendar.DataSource 527
Ektron.Cms.Controls.WebCalendar.DefaultDisplayType 527
Ektron.Cms.Controls.WebCalendar.DisplayTemplatePath 527
Ektron.Cms.Controls.WebCalendar.DisplayType 528
Ektron.Cms.Controls.WebCalendar.DynamicParameter 528
Ektron.Cms.Controls.WebCalendar.Fill 526
Ektron.Cms.Controls.WebCalendar.RequestInformationRef 528
Ektron.Cms.Controls.WebCalendar.Text 529
Ektron.Cms.Controls.WebCalendar.Wrap 526
Ektron.Cms.Controls.WebCalendarForms 548
Ektron.Cms.Controls.WebCalendarForms namespace 548
Classes 549
Ektron.Cms.Controls.WebCalendarForms.AdvancedEditTemplate 549
Ektron.Cms.Controls.WebCalendarForms.AdvancedFormBase 549
Ektron.Cms.Controls.WebCalendarForms.AdvancedInsertTemplate 550
Ektron.Cms.Controls.WebCalendarForms.AdvancedTemplate 550
Ektron.Cms.Controls.WebCalendarForms.AppointmentTemplate 550
Ektron.Cms.Controls.WebCalendarForms.DataKeys 551
Ektron.Cms.Controls.WebCalendarForms.DisplayTemplate 551
Ektron.Cms.Controls.WebCalendarForms.SchedulerStyles 551
Ektron.Cms.Controls.WebSearch 529
Ektron.Cms.Controls.WebSearch.ButtonImgSrc 534
Ektron.Cms.Controls.WebSearch.ButtonText 534
Ektron.Cms.Controls.WebSearch.CustomOrderBy 534
Ektron.Cms.Controls.WebSearch.CustomSearch 535
Ektron.Cms.Controls.WebSearch.DisableForumSearch 535

Ektron.Cms.Controls.WebSearch.DisplayXslt 535
Ektron.Cms.Controls.WebSearch.DynamicContentTemplate 536
Ektron.Cms.Controls.WebSearch.EnableAdvancedLink 536
Ektron.Cms.Controls.WebSearch.EndCount 536
Ektron.Cms.Controls.WebSearch.Fill 530
Ektron.Cms.Controls.WebSearch.FolderID 536
Ektron.Cms.Controls.WebSearch.GetCallbackResult 530
Ektron.Cms.Controls.WebSearch.IsInWorkArea 537
Ektron.Cms.Controls.WebSearch.LinkTarget 537
Ektron.Cms.Controls.WebSearch.LoadSearch 530
Ektron.Cms.Controls.WebSearch.MarkupLanguage 538
Ektron.Cms.Controls.WebSearch.MaxCharacters 538
Ektron.Cms.Controls.WebSearch.MaxTeaserLength 538
Ektron.Cms.Controls.WebSearch.MultipleFolders 539
Ektron.Cms.Controls.WebSearch.OrderBy 539
Ektron.Cms.Controls.WebSearch.OrderDirection 539
Ektron.Cms.Controls.WebSearch.RaiseCallbackEvent 530
Ektron.Cms.Controls.WebSearch.Recursive 540
Ektron.Cms.Controls.WebSearch.RemoveTeaserHtml 540
Ektron.Cms.Controls.WebSearch.ResultCount 540
Ektron.Cms.Controls.WebSearch.ResultDataTable 541
Ektron.Cms.Controls.WebSearch.ResultsPageSize 541
Ektron.Cms.Controls.WebSearch.ResultTagId 542
Ektron.Cms.Controls.WebSearch.ResultXml 542
Ektron.Cms.Controls.WebSearch.SearchFor 542
Ektron.Cms.Controls.WebSearch.SearchSynonyms 543
Ektron.Cms.Controls.WebSearch.SearchText 543
Ektron.Cms.Controls.WebSearch.ShowCategories 544
Ektron.Cms.Controls.WebSearch.ShowCustomSummary 544
Ektron.Cms.Controls.WebSearch.ShowSearchBoxAlways 544
Ektron.Cms.Controls.WebSearch.ShowSearchOptions 545
Ektron.Cms.Controls.WebSearch.ShowSuggestedResults 545
Ektron.Cms.Controls.WebSearch.StartCount 546
Ektron.Cms.Controls.WebSearch.Stylesheet 546
Ektron.Cms.Controls.WebSearch.TaxCategories 546
Ektron.Cms.Controls.WebSearch.TaxonomyOperator 546
Ektron.Cms.Controls.WebSearch.TextBoxSize 547
Ektron.Cms.Controls.WebSearch.TimeTaken 547
Ektron.Cms.Controls.WebSearch.WebSearch 529
Ektron.Cms.Controls.WebSearch.Wrap 531
Ektron.Cms.Controls.WebSearch.XmlDoc 547
Ektron.Cms.DiscussionCategory 1614
Ektron.Cms.DiscussionCategory.BoardID 1614
Ektron.Cms.DiscussionCategory.CategoryID 1615
Ektron.Cms.DiscussionCategory.Name 1615
Ektron.Cms.DiscussionCategory.SortOrder 1615
Ektron.Cms.Extensibility 552
Ektron.Cms.Extensibility namespace 552
Classes 553
Ektron.Cms.Extensibility.ActionRecipientInfo 553
Ektron.Cms.Extensibility.ActionRecipientInfo.ActionRecipientInfo 554
Ektron.Cms.Extensibility.ActionRecipientInfo.DisplayName 554
Ektron.Cms.Extensibility.ActionRecipientInfo.Email 554
Ektron.Cms.Extensibility.ActionRecipientInfo.FirstName 555
Ektron.Cms.Extensibility.ActionRecipientInfo.IsUser 555
Ektron.Cms.Extensibility.ActionRecipientInfo.LastName 555
Ektron.Cms.Extensibility.ActionRecipientInfo.UserId 555
Ektron.Cms.Extensibility.ActionRequestInfo 555
Ektron.Cms.Extensibility.ActionRequestInfo.ActionRequestInfo 556
Ektron.Cms.Extensibility.ActionRequestInfo.ActionRequestType 556
Ektron.Cms.Extensibility.ActionRequestInfo.ActionRequestType enumeration 556
Ektron.Cms.Extensibility.ActionRequestInfo.ErrorMessage 557
Ektron.Cms.Extensibility.ActionRequestInfo.Id 557
Ektron.Cms.Extensibility.ActionRequestInfo.Recipients 557
Ektron.Cms.Extensibility.ActionRequestInfo.SenderId 557
Ektron.Cms.Extensibility.ActionRequestInfo.Type 557
Ektron.Cms.Extensibility.ContentInfo 558
Ektron.Cms.Extensibility.ContentInfo.AssetID 558
Ektron.Cms.Extensibility.ContentInfo.Comments 559
Ektron.Cms.Extensibility.ContentInfo.ErrorMessage 559
Ektron.Cms.Extensibility.ContentInfo.FileAndPath 559
Ektron.Cms.Extensibility.ContentInfo.FileExtension 559
Ektron.Cms.Extensibility.ContentInfo.FileName 559
Ektron.Cms.Extensibility.ContentInfo.FolderID 559
Ektron.Cms.Extensibility.ContentInfo.Html 560

Ektron.Cms.Extensibility.ContentInfo.ID 560
Ektron.Cms.Extensibility.ContentInfo.Language 560
Ektron.Cms.Extensibility.ContentInfo.MimeType 560
Ektron.Cms.Extensibility.ContentInfo.Status 560
Ektron.Cms.Extensibility.ContentInfo.Title 560
Ektron.Cms.Extensibility.ContentInfo.Type 561
Ektron.Cms.Extensibility.ContentInfo.UserID 561
Ektron.Cms.Extensibility.ExtensionAuthor 561
Ektron.Cms.Extensibility.ExtensionAuthor.Author 562
Ektron.Cms.Extensibility.ExtensionAuthor.ExtensionAuthor 561
Ektron.Cms.Extensibility.ExtensionAuthor.ToString 562
Ektron.Cms.Extensibility.ExtensionDescription 562
Ektron.Cms.Extensibility.ExtensionDescription.Desc 563
Ektron.Cms.Extensibility.ExtensionDescription.ExtensionDescription 563
Ektron.Cms.Extensibility.ExtensionDescription.ToString 563
Ektron.Cms.Extensibility.ExtensionEvent 563
Ektron.Cms.Extensibility.ExtensionEvent.Content 580
Ektron.Cms.Extensibility.ExtensionEvent.Folder 581
Ektron.Cms.Extensibility.ExtensionEvent.Form 581
Ektron.Cms.Extensibility.ExtensionEvent.GetActiveDirectoryUserProxy 566
Ektron.Cms.Extensibility.ExtensionEvent.GetAssetProxy 566
Ektron.Cms.Extensibility.ExtensionEvent.GetBlogProxy 566
Ektron.Cms.Extensibility.ExtensionEvent.GetBusinessRulesProxy 567
Ektron.Cms.Extensibility.ExtensionEvent.GetCalendarEventTypeProxy 567
Ektron.Cms.Extensibility.ExtensionEvent.GetCalendarProxy 567
Ektron.Cms.Extensibility.ExtensionEvent.GetCommunityGroupProxy 567
Ektron.Cms.Extensibility.ExtensionEvent.GetContentProxy 567
Ektron.Cms.Extensibility.ExtensionEvent.GetContentRatingProxy 568
Ektron.Cms.Extensibility.ExtensionEvent.GetCustomFieldsProxy 568
Ektron.Cms.Extensibility.ExtensionEvent.GetFavoritesProxy 568
Ektron.Cms.Extensibility.ExtensionEvent.GetFlagProxy 568
Ektron.Cms.Extensibility.ExtensionEvent.GetFolderProxy 569
Ektron.Cms.Extensibility.ExtensionEvent.GetFontProxy 569
Ektron.Cms.Extensibility.ExtensionEvent.GetFormProxy 569
Ektron.Cms.Extensibility.ExtensionEvent.GetFriendsProxy 569
Ektron.Cms.Extensibility.ExtensionEvent.GetLibraryProxy 570
Ektron.Cms.Extensibility.ExtensionEvent.GetMessageBoardProxy 570
Ektron.Cms.Extensibility.ExtensionEvent.GetMetadataProxy 570
Ektron.Cms.Extensibility.ExtensionEvent.GetObjectStateAsXml 570
Ektron.Cms.Extensibility.ExtensionEvent.GetPermissionsProxy 570
Ektron.Cms.Extensibility.ExtensionEvent.GetSearchManagerProxy 571
Ektron.Cms.Extensibility.ExtensionEvent.GetSiteMapProxy 571
Ektron.Cms.Extensibility.ExtensionEvent.GetSiteProxy 571
Ektron.Cms.Extensibility.ExtensionEvent.GetTagsProxy 571
Ektron.Cms.Extensibility.ExtensionEvent.GetTaskCategoryProxy 572
Ektron.Cms.Extensibility.ExtensionEvent.GetTaskCategoryTypeProxy 572
Ektron.Cms.Extensibility.ExtensionEvent.GetTaskProxy 572
Ektron.Cms.Extensibility.ExtensionEvent.GetTaxonomyProxy 572
Ektron.Cms.Extensibility.ExtensionEvent.GetThreadedDiscussionProxy 573
Ektron.Cms.Extensibility.ExtensionEvent.GetUserProxy 573
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterActionRequestSent 573
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentAdd 573
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentDelete 574
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentEdit 574
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentFlagged 574
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFolderAdd 574

Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFolderDelete 574
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFolderEdit 575
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFormSubmit 575
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterPublish 575
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterReviewRejected 575
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserAdd 576
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserDelete 576
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserEdit 576
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserLogin 576
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeActionRequestSent 577
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentAdd 577
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentDelete 577
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentEdit 577
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentFlagged 578
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFolderAdd 578
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFolderDelete 578
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFolderEdit 578
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFormSubmit 578
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforePublish 579
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeReviewRejected 579
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserAdd 579
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserDelete 579
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserEdit 580
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserLogin 580
Ektron.Cms.Extensibility.ExtensionEvent.Site 581
Ektron.Cms.Extensibility.ExtensionEvent.User 581
Ektron.Cms.Extensibility.FlagInfo 581
Ektron.Cms.Extensibility.FlagInfo.ContentHtml 582
Ektron.Cms.Extensibility.FlagInfo.ContentID 582
Ektron.Cms.Extensibility.FlagInfo.ContentLanguage 583
Ektron.Cms.Extensibility.FlagInfo.ContentTitle 583
Ektron.Cms.Extensibility.FlagInfo.EntryId 583
Ektron.Cms.Extensibility.FlagInfo.ErrorMessage 583
Ektron.Cms.Extensibility.FlagInfo.FlagComment 583
Ektron.Cms.Extensibility.FlagInfo.FlagDate 583
Ektron.Cms.Extensibility.FlagInfo.FlagId 584
Ektron.Cms.Extensibility.FlagInfo.FlagName 584
Ektron.Cms.Extensibility.FlagInfo.UserAvatar 584
Ektron.Cms.Extensibility.FlagInfo.UserDisplayName 584
Ektron.Cms.Extensibility.FlagInfo.UserFirstName 584
Ektron.Cms.Extensibility.FlagInfo.UserID 584
Ektron.Cms.Extensibility.FlagInfo.UserLastName 585
Ektron.Cms.Extensibility.FlagInfo.UserName 585
Ektron.Cms.Extensibility.FlagInfo.VisitorID 585
Ektron.Cms.Extensibility.FolderInfo 585
Ektron.Cms.Extensibility.FolderInfo.Description 586
Ektron.Cms.Extensibility.FolderInfo.ErrorMessage 586
Ektron.Cms.Extensibility.FolderInfo.ID 586
Ektron.Cms.Extensibility.FolderInfo.Name 586
Ektron.Cms.Extensibility.FolderInfo.ParentID 586
Ektron.Cms.Extensibility.FolderInfo.ReplicationState 586
Ektron.Cms.Extensibility.FolderInfo.StyleSheet 587
Ektron.Cms.Extensibility.FolderInfo.Type 587
Ektron.Cms.Extensibility.FormField 587
Ektron.Cms.Extensibility.FormField.FieldName 588
Ektron.Cms.Extensibility.FormField.FieldValue 588
Ektron.Cms.Extensibility.FormField.FormField 587
Ektron.Cms.Extensibility.FormInfo 588
Ektron.Cms.Extensibility.FormInfo.ErrorMessage 589
Ektron.Cms.Extensibility.FormInfo.FormDescription 589
Ektron.Cms.Extensibility.FormInfo.FormFields 589
Ektron.Cms.Extensibility.FormInfo.FormTitle 589
Ektron.Cms.Extensibility.FormInfo.this 588
Ektron.Cms.Extensibility.MessageBoardInfo 590
Ektron.Cms.Extensibility.MessageBoardInfo.ErrorMessage 590

Ektron.Cms.Extensibility.MessageBoardInfo.MessageBoardInfo 590
Ektron.Cms.Extensibility.RequestInfo 591
Ektron.Cms.Extensibility.ReviewInfo 591
Ektron.Cms.Extensibility.ReviewInfo.AverageRating 592
Ektron.Cms.Extensibility.ReviewInfo.ContentHtml 592
Ektron.Cms.Extensibility.ReviewInfo.ContentID 592
Ektron.Cms.Extensibility.ReviewInfo.ContentLanguage 592
Ektron.Cms.Extensibility.ReviewInfo.ContentTitle 592
Ektron.Cms.Extensibility.ReviewInfo.ErrorMessage 593
Ektron.Cms.Extensibility.ReviewInfo.ID 593
Ektron.Cms.Extensibility.ReviewInfo.NumberofRatings 593
Ektron.Cms.Extensibility.ReviewInfo.Rating 593
Ektron.Cms.Extensibility.ReviewInfo.RatingDate 593
Ektron.Cms.Extensibility.ReviewInfo.State 593
Ektron.Cms.Extensibility.ReviewInfo.UserAvatar 594
Ektron.Cms.Extensibility.ReviewInfo.UserComments 594
Ektron.Cms.Extensibility.ReviewInfo.UserDisplayName 594
Ektron.Cms.Extensibility.ReviewInfo.UserFirstName 594
Ektron.Cms.Extensibility.ReviewInfo.UserID 594
Ektron.Cms.Extensibility.ReviewInfo.UserLastName 594
Ektron.Cms.Extensibility.ReviewInfo.UserName 595
Ektron.Cms.Extensibility.ReviewInfo.VisitorID 595
Ektron.Cms.Extensibility.SiteInfo 595
Ektron.Cms.Extensibility.SiteInfo.AppPath 595
Ektron.Cms.Extensibility.SiteInfo.IsSSL 596
Ektron.Cms.Extensibility.SiteInfo.Name 596
Ektron.Cms.Extensibility.SiteInfo.PhysicalSitePath 596
Ektron.Cms.Extensibility.SiteInfo.Port 596
Ektron.Cms.Extensibility.SiteInfo.Url 596
Ektron.Cms.Extensibility.UserInfo 596
Ektron.Cms.Extensibility.UserInfo.Addres 597
Ektron.Cms.Extensibility.UserInfo.DisplayName 597
Ektron.Cms.Extensibility.UserInfo.Editor 598
Ektron.Cms.Extensibility.UserInfo.EmailAddress 598
Ektron.Cms.Extensibility.UserInfo.ErrorMessage 598
Ektron.Cms.Extensibility.UserInfo.FirstName 598
Ektron.Cms.Extensibility.UserInfo.ID 598
Ektron.Cms.Extensibility.UserInfo.Language 598
Ektron.Cms.Extensibility.UserInfo.LastName 599
Ektron.Cms.Extensibility.UserInfo.Latitude 599
Ektron.Cms.Extensibility.UserInfo.Longitude 599
Ektron.Cms.Extensibility.UserInfo.Notifications 599
Ektron.Cms.Extensibility.UserInfo.Type 599
Ektron.Cms.Extensibility.UserInfo.UserName 599
Ektron.Cms.FolderRequest 1615
Ektron.Cms.FolderRequest.BreakInheritButton 1617
Ektron.Cms.FolderRequest.CategoryRequired 1618
Ektron.Cms.FolderRequest.ContentSubAssignments 1618
Ektron.Cms.FolderRequest.DomainProduction 1618
Ektron.Cms.FolderRequest.DomainStaging 1618
Ektron.Cms.FolderRequest.EnableReplication 1619
Ektron.Cms.FolderRequest.FileDirectory 1619
Ektron.Cms.FolderRequest.FlagDefIdList 1619
Ektron.Cms.FolderRequest.FlagInherited 1620
Ektron.Cms.FolderRequest.FlagInheritedFrom 1620
Ektron.Cms.FolderRequest.FolderCfldAssignments 1620
Ektron.Cms.FolderRequest.FolderDescription 1621
Ektron.Cms.FolderRequest.FolderId 1621
Ektron.Cms.FolderRequest.FolderName 1621
Ektron.Cms.FolderRequest.FolderType 1621
Ektron.Cms.FolderRequest.ImageDirectory 1622
Ektron.Cms.FolderRequest.IsDomainFolder 1622
Ektron.Cms.FolderRequest.MetaInherited 1622
Ektron.Cms.FolderRequest.MetaInheritedFrom 1623
Ektron.Cms.FolderRequest.ParentId 1623
Ektron.Cms.FolderRequest.PublishActive 1623
Ektron.Cms.FolderRequest.SiteMapPath 1624
Ektron.Cms.FolderRequest.SiteMapPathInherit 1624
Ektron.Cms.FolderRequest.StyleSheet 1624
Ektron.Cms.FolderRequest.SubscriptionProperties 1625
Ektron.Cms.FolderRequest.suppressNotification 1625
Ektron.Cms.FolderRequest.TaxonomyIdList 1625
Ektron.Cms.FolderRequest.TaxonomyInherited 1626
Ektron.Cms.FolderRequest.TaxonomyInheritedFrom 1626
Ektron.Cms.FolderRequest.TemplateFileName 1626
Ektron.Cms.FolderRequest.XmlConfiguration 1627
Ektron.Cms.FolderRequest.XmlInherited 1627
Ektron.Cms.FormFieldDataItem 1627

Ektron.Cms.FormFieldDataItem.DataValue 1628
Ektron.Cms.FormFieldDataItem.FieldName 1628
Ektron.Cms.FormFieldDatalist 1628
Ektron.Cms.FormFieldDatalist.CaptionXPath 1629
Ektron.Cms.FormFieldDatalist.DatalistItems 1629
Ektron.Cms.FormFieldDatalist.DataNamespaces 1629
Ektron.Cms.FormFieldDatalist.DataSelect 1630
Ektron.Cms.FormFieldDatalist.DataSrc 1630
Ektron.Cms.FormFieldDatalist.Language 1630
Ektron.Cms.FormFieldDatalist.Name 1631
Ektron.Cms.FormFieldDatalist.ValueXPath 1631
Ektron.Cms.FormFieldDatalistItem 1631
Ektron.Cms.FormFieldDatalistItem.DataValue 1632
Ektron.Cms.FormFieldDatalistItem.DisplayValue 1632
Ektron.Cms.FormFieldDatalistItem.IsDefault 1632
Ektron.Cms.FormFieldDatalistItem.IsEnabled 1633
Ektron.Cms.FormFieldDefinition 1633
Ektron.Cms.FormFieldDefinition.Basetype 1634
Ektron.Cms.FormFieldDefinition.ContentType 1634
Ektron.Cms.FormFieldDefinition.DatalistName 1634
Ektron.Cms.FormFieldDefinition.Datatype 1635
Ektron.Cms.FormFieldDefinition.DisplayName 1635
Ektron.Cms.FormFieldDefinition.FieldName 1635
Ektron.Cms.FormFieldDefinition.Language 1636
Ektron.Cms.FormFieldDefinition.XPath 1636
Ektron.Cms.FormFieldList 1636
Ektron.Cms.FormFieldList.Datalists 1637
Ektron.Cms.FormFieldList.Fields 1637
Ektron.Cms.FormRequestData 1637
Ektron.Cms.FormRequestData.AutoFill 1638
Ektron.Cms.FormRequestData.FieldList 1639
Ektron.Cms.FormRequestData.Id 1639
Ektron.Cms.FormRequestData.IsPoll 1639
Ektron.Cms.FormRequestData.Language 1639
Ektron.Cms.FormRequestData.PostBackData 1640
Ektron.Cms.FormRequestData.PreFillFielddata 1640
Ektron.Cms.FormRequestData.TagRequired 1640
Ektron.Cms.Framework 995
Ektron.Cms.Framework namespace 995
Classes 1080
Structs, Records, Enums 1082
Ektron.Cms.Framework.Activity 996
Ektron.Cms.Framework.Activity namespace 996
Classes 996
Ektron.Cms.Framework.Activity.Activity 996
Ektron.Cms.Framework.Activity.Activity.Activity 996
Ektron.Cms.Framework.Activity.Activity.Add 997
Ektron.Cms.Framework.Activity.Activity.AddActivityMessage 997
Ektron.Cms.Framework.Activity.Activity.Delete 998
Ektron.Cms.Framework.Activity.Activity.DisableActivityPublishing 998
Ektron.Cms.Framework.Activity.Activity.EnableActivityPublishing 998
Ektron.Cms.Framework.Activity.Activity.GetItem 998
Ektron.Cms.Framework.Activity.Activity.GetList 999
Ektron.Cms.Framework.Activity.Activity.GetListForUser 999
Ektron.Cms.Framework.Activity.Activity.IsActivityPublishingEnabled 1001
Ektron.Cms.Framework.Activity.Activity.Publish 1000
Ektron.Cms.Framework.Activity.Activity.Update 1000
Ektron.Cms.Framework.Activity.Activity.UpdateActivityMessage 1000
Ektron.Cms.Framework.Activity.ActivityQueue 1001
Ektron.Cms.Framework.Activity.ActivityStream 1001
Ektron.Cms.Framework.Activity.ActivityStream.ActivityStream 1001
Ektron.Cms.Framework.Activity.ActivityStream.GetUserActivityStream 1002
Ektron.Cms.Framework.Activity.ActivityStream.SendActivityToAllStreams 1003
Ektron.Cms.Framework.Activity.ActivityStream.SendActivityToStreams 1003
Ektron.Cms.Framework.Activity.ActivityStream.SendActivityToUsersStream 1003
Ektron.Cms.Framework.ActivityType 1004
Ektron.Cms.Framework.ActivityType.Add 1004
Ektron.Cms.Framework.ActivityType.Delete 1004

Ektron.Cms.Framework.ActivityType.GetItem 1005

Ektron.Cms.Framework.ActivityType.GetList 1005

Ektron.Cms.Framework.ActivityType.Update 1005

Ektron.Cms.Framework.Analytics 1005

Ektron.Cms.Framework.Analytics namespace 1005

Classes 1006

Ektron.Cms.Framework.Analytics.ReportAnalytics 1006

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetAbsoluteUniqueVisitorsTrend
1008

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetAdVersions
1009

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetAveragePageViewsTrend
1010

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetBounceRateTrend
1011

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetBrowsers
1012

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetCampaigns
1013

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetCmsSearchTerms
1014

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetConnectionSpeeds
1014, 1015

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetContentbyTitle
1015, 1016

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetContentDetail
1017

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetFlashVersions
1018

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetHostnames
1019

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetJavaSupport
1020

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetKeywords
1021

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetLanguages
1022

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetLocations
1023

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetNetworkLocations
1024

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetNewVsReturningVisitors
1025

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetOperatingSystems
1026, 1027

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetPageViewsTrend
1027, 1028

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetPlatforms
1028, 1029

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetScreenColors
1029, 1030

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetScreenResolutions
1030, 1031

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetTimeOnSiteTrend
1031, 1032

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetTopContent
1032, 1033

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetTopExitPages
1034

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetTopLandingPages
1035

Ektron.Cms.Framework.Analytics.ReportAnalytics.GetVisitsTrend
1036

Ektron.Cms.Framework.Analytics.ReportAnalytics.ReportAnalytics
1006

Ektron.Cms.Framework.Analytics.SearchAnalytics 1036

Ektron.Cms.Framework.Analytics.SearchAnalytics.SearchAnalytics
1037

Ektron.Cms.Framework.ApiAccessMode 1082

Ektron.Cms.Framework.ApiAccessMode enumeration 1082

Ektron.Cms.Framework.Calendar 1037

Ektron.Cms.Framework.Calendar namespace 1037

Classes 1037
Ektron.Cms.Framework.Calendar.WebCalendar 1037
Ektron.Cms.Framework.Calendar.WebCalendar.Add 1039
Ektron.Cms.Framework.Calendar.WebCalendar.Delete 1039
Ektron.Cms.Framework.Calendar.WebCalendar.GetCalendar 1039
Ektron.Cms.Framework.Calendar.WebCalendar.GetICalendarFeed 1039
Ektron.Cms.Framework.Calendar.WebCalendar.GetPublicCalendar 1039
Ektron.Cms.Framework.Calendar.WebCalendar.Update 1040
Ektron.Cms.Framework.Calendar.WebCalendar.WebCalendar 1038
Ektron.Cms.Framework.Calendar.WebEvent 1040
Ektron.Cms.Framework.Calendar.WebEvent.Add 1041
Ektron.Cms.Framework.Calendar.WebEvent.CancelOccurrence 1042
Ektron.Cms.Framework.Calendar.WebEvent.CreateVariance 1042, 1043
Ektron.Cms.Framework.Calendar.WebEvent.Delete 1043
Ektron.Cms.Framework.Calendar.WebEvent.GetEventOccurrenceList 1043
Ektron.Cms.Framework.Calendar.WebEvent.GetItem 1044
Ektron.Cms.Framework.Calendar.WebEvent.GetList 1044, 1045
Ektron.Cms.Framework.Calendar.WebEvent.GetNonVariantEventList 1045, 1046
Ektron.Cms.Framework.Calendar.WebEvent.GetVarianceEventList 1046
Ektron.Cms.Framework.Calendar.WebEvent.Update 1047
Ektron.Cms.Framework.Calendar.WebEvent.WebEvent 1041
Ektron.Cms.Framework.CmsApi 1080
Ektron.Cms.Framework.CmsApi.ApiMode 1081
Ektron.Cms.Framework.CmsApi.CmsApi 1080, 1081
Ektron.Cms.Framework.CmsApi.ContentLanguage 1081
Ektron.Cms.Framework.CmsApi.InPreviewMode 1081
Ektron.Cms.Framework.CmsApi.RequestInformation 1082
Ektron.Cms.Framework.CmsApi.SitePath 1082
Ektron.Cms.Framework.CmsApi.UserId 1082
Ektron.Cms.Framework.Core 1047
Ektron.Cms.Framework.Core namespace 1047
Ektron.Cms.Framework.Core.Content 1047
Ektron.Cms.Framework.Core.Content namespace 1047
Classes 1047
Ektron.Cms.Framework.Core.Content.Content 1048
Ektron.Cms.Framework.Core.Content.Content.Add 1048
Ektron.Cms.Framework.Core.Content.Content.Content 1048
Ektron.Cms.Framework.Core.Content.Content.Delete 1049
Ektron.Cms.Framework.Core.Content.Content.GetItem 1049
Ektron.Cms.Framework.Core.Content.Content.GetList 1049
Ektron.Cms.Framework.Core.Content.Content.Update 1049
Ektron.Cms.Framework.Core.Folder 1050
Ektron.Cms.Framework.Core.Folder namespace 1050
Classes 1050
Ektron.Cms.Framework.Core.Folder.Folder 1050
Ektron.Cms.Framework.Core.Folder.Folder.Add 1051
Ektron.Cms.Framework.Core.Folder.Folder.Delete 1051
Ektron.Cms.Framework.Core.Folder.Folder.Folder 1050
Ektron.Cms.Framework.Core.Folder.Folder.GetItem 1051
Ektron.Cms.Framework.Core.Folder.Folder.GetList 1051
Ektron.Cms.Framework.Core.Folder.Folder.Update 1052
Ektron.Cms.Framework.Messaging 1052
Ektron.Cms.Framework.Messaging namespace 1052
Classes 1052
Ektron.Cms.Framework.Messaging.CmsMessage 1052
Ektron.Cms.Framework.Messaging.CmsMessage._messageService 1053
Ektron.Cms.Framework.Messaging.CmsMessage.Add 1054
Ektron.Cms.Framework.Messaging.CmsMessage.CmsMessage 1053
Ektron.Cms.Framework.Messaging.CmsMessage.Delete 1054
Ektron.Cms.Framework.Messaging.CmsMessage.GetDefaultItemByType 1054
Ektron.Cms.Framework.Messaging.CmsMessage.GetItem 1055
Ektron.Cms.Framework.Messaging.CmsMessage.GetItemCollection 1055
Ektron.Cms.Framework.Messaging.CmsMessage.GetList 1055
Ektron.Cms.Framework.Messaging.CmsMessage.MessageServi

ce
1056

Ektron.Cms.Framework.Messaging.CmsMessage.Update 1055

Ektron.Cms.Framework.Messaging.CmsMessageType 1056

Ektron.Cms.Framework.Messaging.CmsMessageType._messageTypeService
1057

Ektron.Cms.Framework.Messaging.CmsMessageType.Add
1058

Ektron.Cms.Framework.Messaging.CmsMessageType.CmsMessageType
1057

Ektron.Cms.Framework.Messaging.CmsMessageType.Delete
1058

Ektron.Cms.Framework.Messaging.CmsMessageType.GetItem
1058

Ektron.Cms.Framework.Messaging.CmsMessageType.GetList
1058

Ektron.Cms.Framework.Messaging.CmsMessageType.GetTokenList
1059

Ektron.Cms.Framework.Messaging.CmsMessageType.MessageTypeService
1059

Ektron.Cms.Framework.Messaging.CmsMessageType.Update
1059

Ektron.Cms.Framework.Notifications 1060

Ektron.Cms.Framework.Notifications namespace 1060

Classes 1060

Ektron.Cms.Framework.Notifications.Notification 1060

Ektron.Cms.Framework.Notifications.Notification.Notification
1060

Ektron.Cms.Framework.Notifications.Notification.Send 1061

Ektron.Cms.Framework.Notifications.NotificationAgentSetting
1062

Ektron.Cms.Framework.Notifications.NotificationAgentSetting.Add
1062

Ektron.Cms.Framework.Notifications.NotificationAgentSetting.Delete
1063

Ektron.Cms.Framework.Notifications.NotificationAgentSetting.GetAgent
1063

Ektron.Cms.Framework.Notifications.NotificationAgentSetting.GetItem
1063

Ektron.Cms.Framework.Notifications.NotificationAgentSetting.GetList
1063

Ektron.Cms.Framework.Notifications.NotificationAgentSetting.GetRegisteredAgentList
1064

Ektron.Cms.Framework.Notifications.NotificationAgentSetting.NotificationAgentSetting
1062

Ektron.Cms.Framework.Notifications.NotificationAgentSetting.Update
1064

Ektron.Cms.Framework.Notifications.NotificationPreference
1064

Ektron.Cms.Framework.Notifications.NotificationPreference.Add
1065

Ektron.Cms.Framework.Notifications.NotificationPreference.Delete
1065

Ektron.Cms.Framework.Notifications.NotificationPreference.GetDefaultPreferenceList
1065

Ektron.Cms.Framework.Notifications.NotificationPreference.GetItem
1066

Ektron.Cms.Framework.Notifications.NotificationPreference.GetList
1066

Ektron.Cms.Framework.Notifications.NotificationPreference.GetUserNotificationListForActivity
1066

Ektron.Cms.Framework.Notifications.NotificationPreference.NotificationPreference
1064

Ektron.Cms.Framework.Notifications.NotificationPreference.SaveUserPreferences
1066

Ektron.Cms.Framework.Notifications.NotificationPreference.Update
1067

Ektron.Cms.Framework.Notifications.UserNotificationSetting
1067

Ektron.Cms.Framework.Notifications.UserNotificationSetting.Add
1068

Ektron.Cms.Framework.Notifications.UserNotificationSetting.Delete
1068

Ektron.Cms.Framework.Notifications.UserNotificationSetting.GetItem
1068

Ektron.Cms.Framework.Notifications.UserNotificationSetting.GetList 1069

Ektron.Cms.Framework.Notifications.UserNotificationSetting.Update 1069

Ektron.Cms.Framework.Notifications.UserNotificationSetting.UserNotificationSetting 1067

Ektron.Cms.Framework.Notifications.UserNotificationSetting.VerifyValidationCode 1069

Ektron.Cms.Framework.SocialNetworking 1069

Ektron.Cms.Framework.SocialNetworking namespace 1069

Classes 1070

Ektron.Cms.Framework.SocialNetworking.MicroMessage 1070

Ektron.Cms.Framework.SocialNetworking.MicroMessage._MicroMessageBL 1071

Ektron.Cms.Framework.SocialNetworking.MicroMessage.Add 1072

Ektron.Cms.Framework.SocialNetworking.MicroMessage.AddReply 1072

Ektron.Cms.Framework.SocialNetworking.MicroMessage.Delete 1072

Ektron.Cms.Framework.SocialNetworking.MicroMessage.GetColleagueMessageList 1073

Ektron.Cms.Framework.SocialNetworking.MicroMessage.GetInstance 1076

Ektron.Cms.Framework.SocialNetworking.MicroMessage.GetItem 1073

Ektron.Cms.Framework.SocialNetworking.MicroMessage.GetList 1073

Ektron.Cms.Framework.SocialNetworking.MicroMessage.GetReplyList 1074

Ektron.Cms.Framework.SocialNetworking.MicroMessage.GetSearchList 1074

Ektron.Cms.Framework.SocialNetworking.MicroMessage.GetTimeLineMessageList 1074, 1075

Ektron.Cms.Framework.SocialNetworking.MicroMessage.GetUserMessageList 1075

Ektron.Cms.Framework.SocialNetworking.MicroMessage.IsFullTextSearchInstalled 1075

Ektron.Cms.Framework.SocialNetworking.MicroMessage.IsSpam 1075

Ektron.Cms.Framework.SocialNetworking.MicroMessage.MicroMessage 1070, 1071

Ektron.Cms.Framework.Users 1076

Ektron.Cms.Framework.Users namespace 1076

Classes 1076

Ektron.Cms.Framework.Users.User 1076

Ektron.Cms.Framework.Users.User._userService 1078

Ektron.Cms.Framework.Users.User.Add 1078

Ektron.Cms.Framework.Users.User.Delete 1078

Ektron.Cms.Framework.Users.User.GetItem 1078

Ektron.Cms.Framework.Users.User.GetList 1079

Ektron.Cms.Framework.Users.User.Update 1079

Ektron.Cms.Framework.Users.User.User 1077

Ektron.Cms.Framework.Users.User.UserService 1079

Ektron.Cms.ImageMediaData 1506

Ektron.Cms.ImageMediaData.Thumbnails 1506

Ektron.Cms.MediaData 1510

Ektron.Cms.MediaData.Alt 1511

Ektron.Cms.MediaData.FileName 1511

Ektron.Cms.MediaData.FilePath 1512

Ektron.Cms.MediaData.Height 1512

Ektron.Cms.MediaData.Id 1512

Ektron.Cms.MediaData.MimeType 1512

Ektron.Cms.MediaData.Sequence 1512

Ektron.Cms.MediaData.Title 1513

Ektron.Cms.MediaData.Width 1513

Ektron.Cms.MediaGalleryData 1513

Ektron.Cms.MediaGalleryData.Images 1514

Ektron.Cms.MediaGalleryData.ObjectId 1514

Ektron.Cms.PagingInfo 1641

Ektron.Cms.PagingInfo.CurrentPage 1641

Ektron.Cms.PagingInfo.EndRow 1642

Ektron.Cms.PagingInfo.RecordsPerPage 1642

Ektron.Cms.PagingInfo.StartRow 1642
Ektron.Cms.PagingInfo.TotalPages 1642
Ektron.Cms.PagingInfo.TotalRecords 1642
Ektron.Cms.TaxonomyRequest 1643
Ektron.Cms.TaxonomyRequest.Depth 1644
Ektron.Cms.TaxonomyRequest.FolderId 1644
Ektron.Cms.TaxonomyRequest.IncludeItems 1644
Ektron.Cms.TaxonomyRequest.Preview 1645
Ektron.Cms.TaxonomyRequest.ReadCount 1645
Ektron.Cms.TaxonomyRequest.TaxonomyIdList 1645
Ektron.Cms.TaxonomyRequest.TaxonomyItemType 1645
Ektron.Cms.TaxonomyRequest.TaxonomyPath 1646
Ektron.Cms.ThumbnailData 1568
Ektron.Cms.ThumbnailData.Height 1568
Ektron.Cms.ThumbnailData.Width 1568
Ektron.Cms.ThumbnailDefaultData 1569
Ektron.Cms.ThumbnailDefaultData.FileNameFormat 1569
Ektron.Cms.ThumbnailDefaultData.Height 1569
Ektron.Cms.ThumbnailDefaultData.Id 1570
Ektron.Cms.ThumbnailDefaultData.MimeType 1570
Ektron.Cms.ThumbnailDefaultData.Title 1570
Ektron.Cms.ThumbnailDefaultData.Width 1570
Ektron.Cms.UriFavorite 1646
Ektron.Cms.UriFavorite.FavoriteDesc 1647
Ektron.Cms.UriFavorite.FavoriteId 1647
Ektron.Cms.UriFavorite.FavoriteName 1647
Ektron.Cms.UriFavorite.FavoriteUri 1648
Ektron.Cms.UriFavorite.UserId 1648
Ektron.Cms.WorkSpaceGroupRequest 1648
Ektron.Cms.WorkSpaceGroupRequest.GroupId 1649
Ektron.Cms.WorkSpaceUserRequest 1649
Ektron.Cms.WorkSpaceUserRequest.UserId 1650
Ektron.Services 1083
Ektron.Services namespace 1083
Classes 1101
Ektron.Services.ActiveDirectoryUser 1103
Ektron.Services.ActiveDirectoryUser.AddADGroupToCMS 1104
Ektron.Services.ActiveDirectoryUser.DeSynchUserGroups 1104
Ektron.Services.ActiveDirectoryUser.DeSynchUsers 1105
Ektron.Services.ActiveDirectoryUser.GetADMapping 1105
Ektron.Services.ActiveDirectoryUser.GetADStatus 1106
Ektron.Services.ActiveDirectoryUser.GetAvailableADGroups 1106
Ektron.Services.ActiveDirectoryUser.GetAvailableADUsers 1107
Ektron.Services.ActiveDirectoryUser.GetCMSGroupsToSync 1108
Ektron.Services.ActiveDirectoryUser.GetCMSRelationshipsToSync 1108
Ektron.Services.ActiveDirectoryUser.GetCMSUsersToSync 1109
Ektron.Services.ActiveDirectoryUser.GetDomains 1109
Ektron.Services.ActiveDirectoryUser.MapCMSUserGroupToAD 1110
Ektron.Services.ActiveDirectoryUser.MapCMSUserToAD 1110
Ektron.Services.ActiveDirectoryUser.UpdateADMapping 1111
Ektron.Services.ActiveDirectoryUser.UpdateADUsersGroups 1111
Ektron.Services.ActiveDirectoryUser.UpdateUsersInfoFromAD 1112
Ektron.Services.Analytics 1133
Ektron.Services.Analytics.GetAbsoluteUniqueVisitorsTrend 1135, 1136
Ektron.Services.Analytics.GetAdVersions 1137
Ektron.Services.Analytics.GetAllTrafficSources 1138, 1139
Ektron.Services.Analytics.GetAveragePageViewsTrend 1140
Ektron.Services.Analytics.GetBounceRateTrend 1141
Ektron.Services.Analytics.GetBrowsers 1142, 1143
Ektron.Services.Analytics.GetCampaigns 1144
Ektron.Services.Analytics.GetCmsSearchTerms 1145
Ektron.Services.Analytics.GetConnectionSpeeds 1146
Ektron.Services.Analytics.GetContentbyTitle 1147, 1148
Ektron.Services.Analytics.GetContentDetail 1149
Ektron.Services.Analytics.GetDirectTraffic 1150, 1151
Ektron.Services.Analytics.GetFlashVersions 1151, 1152
Ektron.Services.Analytics.GetHostnames 1153
Ektron.Services.Analytics.GetJavaSupport 1154, 1155
Ektron.Services.Analytics.GetKeywords 1156
Ektron.Services.Analytics.GetLanguages 1157, 1158
Ektron.Services.Analytics.GetLocations 1159
Ektron.Services.Analytics.GetNetworkLocations 1160, 1161

Ektron.Services.Analytics.GetNewVsReturningVisitors 1161, 1162

Ektron.Services.Analytics.GetOperatingSystems 1163

Ektron.Services.Analytics.GetPageViewsTrend 1164, 1165

Ektron.Services.Analytics.GetPlatforms 1166

Ektron.Services.Analytics.GetReferringSites 1167, 1168

Ektron.Services.Analytics.GetScreenColors 1168, 1169

Ektron.Services.Analytics.GetScreenResolutions 1170

Ektron.Services.Analytics.GetSearchEngines 1171, 1172

Ektron.Services.Analytics.GetTimeOnSiteTrend 1173

Ektron.Services.Analytics.GetTopContent 1174, 1175

Ektron.Services.Analytics.GetTopExitPages 1176

Ektron.Services.Analytics.GetTopLandingPages 1177, 1178

Ektron.Services.Analytics.GetUserDefinedValue 1178, 1179

Ektron.Services.Analytics.GetVisitsTrend 1180

Ektron.Services.Asset 1112

Ektron.Services.Asset.GetAssetMgtConfigInfo 1113

Ektron.Services.Asset.GetAssetMgtInfo 1113

Ektron.Services.Asset.GetAssetSupertypes 1114

Ektron.Services.Asset.GetEditSnippet 1114

Ektron.Services.Asset.GetMyAssetMgtID 1115

Ektron.Services.Asset.MakeCopyDMSAsset 1115

Ektron.Services.Asset.SetAssetMgtConfigInfo 1115

Ektron.Services.Blog 1116

Ektron.Services.Blog.AddBlog 1117

Ektron.Services.Blog.AddCommentForPost 1118

Ektron.Services.Blog.AddPost 1119

Ektron.Services.Blog.DeleteBlog 1119

Ektron.Services.Blog.DeleteBlogPostComment 1120

Ektron.Services.Blog.DeletePost 1120

Ektron.Services.Blog.GetBlankBlogPostData 1120

Ektron.Services.Blog.GetBlog 1121, 1122

Ektron.Services.Blog.GetBlogData 1123, 1124, 1125

Ektron.Services.Blog.GetBlogDataForPost 1126

Ektron.Services.Blog.GetBlogPostData 1127

Ektron.Services.Blog.GetBlogRoll 1128

Ektron.Services.Blog.GetPost 1129

Ektron.Services.Blog.GetPostbyID 1129

Ektron.Services.Blog.GetUserBlog 1130

Ektron.Services.Blog.UpdateBlogProperties 1130

Ektron.Services.BusinessRules 1130

Ektron.Services.BusinessRules.GetBusinessRules 1132

Ektron.Services.Calendar 1133

Ektron.Services.CommunityGroup 1181

Ektron.Services.CommunityGroup.AddCommunityGroup 1182

Ektron.Services.CommunityGroup.AddUserToCommunityGroup 1183, 1184

Ektron.Services.CommunityGroup.ApprovePendingGroupUser 1184

Ektron.Services.CommunityGroup.CancelJoinRequestForCommunityGroup 1185

Ektron.Services.CommunityGroup.DeleteCommunityGroupByID 1185

Ektron.Services.CommunityGroup.DeletePendingGroupUser 1186

Ektron.Services.CommunityGroup.GetAllCommunityGroups 1186

Ektron.Services.CommunityGroup.GetAllCommunityGroupsInDirectory 1186

Ektron.Services.CommunityGroup.GetAllUnassignedCommunityGroupUsers 1187

Ektron.Services.CommunityGroup.GetCommunityGroupBlogIds 1187

Ektron.Services.CommunityGroup.GetCommunityGroupByID 1188

Ektron.Services.CommunityGroup.GetCommunityGroupDiscussionBoardIds 1188

Ektron.Services.CommunityGroup.GetCommunityGroups 1189

Ektron.Services.CommunityGroup.GetCommunityGroupUserIDs 1189

Ektron.Services.CommunityGroup.GetCommunityGroupUsers 1190, 1191

Ektron.Services.CommunityGroup.GetFolderIdForCommunityGroup 1191

Ektron.Services.CommunityGroup.GetGroupMemberStatus 1192

Ektron.Services.CommunityGroup.GetInvitedCommunityGroups 1192

Ektron.Services.CommunityGroup.GetMostPopularCommunityGroups 1193, 1194

Ektron.Services.CommunityGroup.GetPendingCommunityGroup

s
1194

Ektron.Services.CommunityGroup.GetPendingCommunityGroup Users
1195

Ektron.Services.CommunityGroup.InviteUsers 1195

Ektron.Services.CommunityGroup.IsCommunityGroupAdmin
1196

Ektron.Services.CommunityGroup.IsUserInCommunityGroup
1196

Ektron.Services.CommunityGroup.RemoveUserFromCommunity Group
1197

Ektron.Services.CommunityGroup.UpdateCommunityGroup
1197

Ektron.Services.Content 1198

Ektron.Services.Content.AddContent 1200, 1201, 1202

Ektron.Services.Content.CheckOutContent 1203

Ektron.Services.Content.CopyContentToFolder 1203

Ektron.Services.Content.DeleteContentItem 1204

Ektron.Services.Content.GetAddViewLanguage 1204

Ektron.Services.Content.GetAllComments 1205

Ektron.Services.Content.GetAllTemplates 1205

Ektron.Services.Content.GetAllUnassignedItemApprovals 1206

Ektron.Services.Content.GetChildContent 1206, 1207

Ektron.Services.Content.GetContent 1208

Ektron.Services.Content.GetContentByHistory 1208

Ektron.Services.Content.GetContentForEditing 1209

Ektron.Services.Content.GetContentState 1209

Ektron.Services.Content.GetContentStatus 1210

Ektron.Services.Content.GetDomain 1210

Ektron.Services.Content.MoveContentToFolder 1210

Ektron.Services.Content.PublishContent 1211

Ektron.Services.Content.RenameContent 1211

Ektron.Services.Content.SaveContent 1212

Ektron.Services.Content.SubmitForDelete 1213

Ektron.Services.Content.UndoCheckout 1213

Ektron.Services.Content.UpdateContentMetaData 1213

Ektron.Services.ContentRating 1214

Ektron.Services.ContentRating.AddContentRating 1215

Ektron.Services.ContentRating.GetContentRatingResults 1215

Ektron.Services.ContentRating.PurgeContentRatings 1216

Ektron.Services.CustomFields 1216

Ektron.Services.CustomFields.AddNewEditableCustomFieldAssi gnments
1218

Ektron.Services.CustomFields.BoolToYesNo 1218

Ektron.Services.CustomFields.GetEditableCustomFieldAssignm ents
1219

Ektron.Services.CustomFields.GetEnhancedMetadataArea 1220

Ektron.Services.CustomFields.GetEnhancedMetadataScript
1220

Ektron.Services.CustomFields.GetFieldsByFolder 1220

Ektron.Services.CustomFields.GetSearchProperties 1221

Ektron.Services.CustomFields.ProcessCustomFields 1221

Ektron.Services.CustomFields.SearchAllAssets 1222

Ektron.Services.CustomFields.WriteFilteredMetadataForEdit
1222

Ektron.Services.CustomFields.WriteFilteredMetadataForView
1223

Ektron.Services.CustomFields.WriteLibrarySearchExtended
1223

Ektron.Services.CustomFields.WriteMetadataDefaultForEdit
1224

Ektron.Services.CustomFields.WriteMetadataTypeForView 1224

Ektron.Services.CustomFields.WritePublicSearch 1225

Ektron.Services.Favorites 1225

Ektron.Services.Favorites.AddContentFavorite 1226, 1227

Ektron.Services.Favorites.DeleteContentFavorite 1227

Ektron.Services.Favorites.GetContentFavorites 1228, 1229

Ektron.Services.Favorites.GetFavoritesDirectory 1230

Ektron.Services.Favorites.IsContentFavorite 1230

Ektron.Services.Favorites.MoveContentFavorite 1231

Ektron.Services.Flag 1231

Ektron.Services.Flag.AddFlagObject 1232

Ektron.Services.Flag.AssignFlagToContent 1233

Ektron.Services.Flag.GetAllFlagEntries 1233

Ektron.Services.Flag.GetContentFlagData 1234

Ektron.Services.Flag.GetFlagDataById 1234

Ektron.Services.Flag.GetFlaggedRank 1235

Ektron.Services.Flag.PurgeFlagEntries 1235

Ektron.Services.Flag.RemoveObjectObject 1235

Ektron.Services.Flag.UpdateFlagObject 1236

Ektron.Services.Folder 1236
Ektron.Services.Folder.AddFolder 1237
Ektron.Services.Folder.CreateFolder 1238
Ektron.Services.Folder.DeleteFolder 1238
Ektron.Services.Folder.GetChildFolders 1239
Ektron.Services.Folder.GetFolder 1240, 1241
Ektron.Services.Folder.GetFolderID 1241, 1242
Ektron.Services.Folder.GetPath 1242
Ektron.Services.Folder.RenameFolder 1243
Ektron.Services.Folder.UpdateFolder 1243, 1244
Ektron.Services.Font 1244
Ektron.Services.Font.AddFont 1245
Ektron.Services.Font.DeleteFont 1245
Ektron.Services.Font.GetAllFonts 1246
Ektron.Services.Font.GetFont 1246
Ektron.Services.Font.GetFontConfigList 1247
Ektron.Services.Font.UpdateFont 1247
Ektron.Services.Form 1247
Ektron.Services.Form.AssignFormTask 1248
Ektron.Services.Form.EmailFormFieldData 1249
Ektron.Services.Form.GetAllForms 1250
Ektron.Services.Form.GetForm 1251
Ektron.Services.Form.GetFormFieldData 1252
Ektron.Services.Form.GetFormFieldList 1253
Ektron.Services.Form.GetFormFieldQuestions 1253
Ektron.Services.Form.GetFormTitle 1254
Ektron.Services.Form.SubmitFormFieldData 1254
Ektron.Services.Form.UpdateFormFieldData 1255
Ektron.Services.Friends 1255
Ektron.Services.Friends.AcceptPendingFriend 1257
Ektron.Services.Friends.AddFriend 1257, 1258
Ektron.Services.Friends.AddPendingFriend 1258
Ektron.Services.Friends.AddSelectedFriend 1259
Ektron.Services.Friends.DeleteFriend 1259
Ektron.Services.Friends.DeletePendingFriendRequest 1260
Ektron.Services.Friends.DeleteSelectedFriend 1261
Ektron.Services.Friends.GetFriends 1262, 1263
Ektron.Services.Friends.GetFriendsDirectory 1264
Ektron.Services.Friends.GetFriendStatus 1264
Ektron.Services.Friends.GetInvitedUsers 1265
Ektron.Services.Friends.GetPendingFriends 1266
Ektron.Services.Friends.GetSelectedFriends 1266
Ektron.Services.Friends.GetSentFriendInvitations 1267
Ektron.Services.Friends.InviteUsers 1267
Ektron.Services.Friends.IsFriend 1268
Ektron.Services.Friends.IsSelectedFriend 1268
Ektron.Services.Friends.MoveFriend 1269
Ektron.Services.Library 1269
Ektron.Services.Library.AddLoadBalanceItem 1270
Ektron.Services.Library.DeleteLibraryItem 1271
Ektron.Services.Library.DeleteLibraryItemById 1271
Ektron.Services.Library.DeleteLoadBalanceItem 1272
Ektron.Services.Library.GetAllChildLibItems 1272
Ektron.Services.Library.GetAllLoadBalancePaths 1273, 1274
Ektron.Services.Library.GetLibraryItem 1274, 1275
Ektron.Services.Library.GetLibrarySettings 1275
Ektron.Services.Library.GetLibraryTypes 1276
Ektron.Services.Library.GetLoadBalancePath 1276
Ektron.Services.Library.InternalLibrarySearch 1277
Ektron.Services.Library.UpdateLibrary 1277
Ektron.Services.Library.UpdateLibrarySettings 1278
Ektron.Services.Library.UpdateLoadBalanceSettings 1278
Ektron.Services.Library.UpdateQlinkTemplates 1278
Ektron.Services.MessageBoard 1279
Ektron.Services.MessageBoard.AddMessageBoardEntry 1280
Ektron.Services.MessageBoard.ApproveMessage 1281, 1282
Ektron.Services.MessageBoard.DeleteMessageBoardEntries 1283, 1284
Ektron.Services.MessageBoard.DeleteMessageBoardEntriesBy User 1285, 1286, 1287
Ektron.Services.MessageBoard.DeleteMessageBoardEntry 1288, 1289
Ektron.Services.MessageBoard.GetAllMessageBoardEntries 1290, 1291
Ektron.Services.MessageBoard.GetAllMessageBoardEntriesBy User 1293, 1294
Ektron.Services.MessageBoard.GetMessageBoardEntry 1295, 1296
Ektron.Services.MessageBoard.IsUserMessageBoardAdmin

1297
Ektron.Services.MessageBoard.UpdateMessageText 1298, 1299
Ektron.Services.Metadata 1300
Ektron.Services.Metadata.AddMetaDataType 1301
Ektron.Services.Metadata.DeleteMetadataType 1301
Ektron.Services.Metadata.GetContentMetadataList 1302
Ektron.Services.Metadata.GetMetadataType 1303
Ektron.Services.Metadata.GetMetaDataTypes 1303
Ektron.Services.Metadata.UpdateMetadataType 1304
Ektron.Services.Permissions 1304
Ektron.Services.Permissions.AddItemPermission 1305
Ektron.Services.Permissions.DeleteItemPermission 1306
Ektron.Services.Permissions.DisableContentInheritance 1306
Ektron.Services.Permissions.DisableFolderInheritance 1307
Ektron.Services.Permissions.EnableContentInheritance 1307
Ektron.Services.Permissions.EnableFolderInheritance 1308
Ektron.Services.Permissions.GetUserPermissions 1309
Ektron.Services.Permissions.LoadPermissions 1310
Ektron.Services.Permissions.UpdateItemPermission 1310
Ektron.Services.SearchManager 1311
Ektron.Services.SearchManager.ExecuteSearch 1312
Ektron.Services.SearchManager.Search 1313, 1314
Ektron.Services.SearchManager.SearchForCategories 1315
Ektron.Services.SearchManager.SearchTable 1315
Ektron.Services.Site 1316
Ektron.Services.Site.BreakTaskPermissionInherit 1317
Ektron.Services.Site.GetAllActiveLanguages 1317
Ektron.Services.Site.GetAllLanguages 1318
Ektron.Services.Site.GetClientDayNames 1319
Ektron.Services.Site.GetClientMonthNames 1319
Ektron.Services.Site.GetCookieValue 1320
Ektron.Services.Site.GetLanguage 1320
Ektron.Services.Site.GetLanguages 1321
Ektron.Services.Site.GetSiteVariables 1321
Ektron.Services.Site.GetTaskPermission 1322
Ektron.Services.Site.IsTaskPermissionInherited 1322
Ektron.Services.Site.SetCookieValue 1323
Ektron.Services.Site.SetTaskPermission 1323
Ektron.Services.Site.UpdateSiteEnabledLanguages 1324
Ektron.Services.SiteMap 1324
Ektron.Services.SiteMap.GetFolderBreadcrumbPath 1325
Ektron.Services.SiteMap.GetWebsiteMap 1325
Ektron.Services.SiteMap.RenderFolderBreadcrumbHtml 1326
Ektron.Services.SiteMap.RenderWebsiteMap 1327
Ektron.Services.Tags 1327
Ektron.Services.Tags.AddTag 1329
Ektron.Services.Tags.AddTagToCommunityGroup 1330, 1331
Ektron.Services.Tags.AddTagToContent 1332, 1333
Ektron.Services.Tags.AddTagToObject 1334
Ektron.Services.Tags.AddTagToUser 1336, 1337
Ektron.Services.Tags.DeleteTagById 1337
Ektron.Services.Tags.DeleteTagOnObject 1338
Ektron.Services.Tags.DeleteTagsByUser 1338
Ektron.Services.Tags.DeleteTagsOnCommunityGroup 1339
Ektron.Services.Tags.DeleteTagsOnContent 1339
Ektron.Services.Tags.DeleteTagsOnObject 1340
Ektron.Services.Tags.DeleteTagsOnUser 1340
Ektron.Services.Tags.EditTag 1341
Ektron.Services.Tags.GetAllTagRelatedContent 1341
Ektron.Services.Tags.GetAllTagRelatedUsers 1342
Ektron.Services.Tags.GetAllTags 1343
Ektron.Services.Tags.GetDefaultTags 1344, 1345
Ektron.Services.Tags.GetTagByID 1345
Ektron.Services.Tags.GetTagByName 1346
Ektron.Services.Tags.GetTagCloud 1346
Ektron.Services.Tags.GetTagsForContent 1347
Ektron.Services.Tags.GetTagsForObject 1347, 1348
Ektron.Services.Tags.GetTagsForUser 1349
Ektron.Services.Tags.RemoveTagAsDefault 1350
Ektron.Services.Tags.SaveTagAsDefault 1350
Ektron.Services.Task 1085
Ektron.Services.Task namespace 1085
Classes 1085
Ektron.Services.Task.Task 1085
Ektron.Services.Task.Task.AddTask 1086
Ektron.Services.Task.Task.AssignTaskToUser 1087
Ektron.Services.Task.Task.AssignTaskToUsergroup 1088
Ektron.Services.Task.Task.DeleteTask 1089

Ektron.Services.Task.Task.DeleteTasks 1090
Ektron.Services.Task.Task.GetTask 1091
Ektron.Services.Task.Task.GetTasks 1091
Ektron.Services.Task.Task.GetTasksAssignedByUser 1092
Ektron.Services.Task.Task.GetTasksAssignedToUser 1093
Ektron.Services.Task.Task.GetTasksAssignedToUsergroup 1094, 1095
Ektron.Services.Task.Task.GetTaskState 1095
Ektron.Services.Task.Task.SetStateForTask 1095
Ektron.Services.Task.Task.UpdateTask 1096
Ektron.Services.Task.TaskCategory 1096
Ektron.Services.Task.TaskCategory.AddTaskCategory 1097
Ektron.Services.Task.TaskCategory.DeleteTaskCategory 1097
Ektron.Services.Task.TaskCategory.GetAllCategories 1098
Ektron.Services.Task.TaskCategory.GetTaskCategory 1098
Ektron.Services.Task.TaskCategory.UpdateTaskCategory 1099
Ektron.Services.Task.TaskCategoryType 1099
Ektron.Services.Task.TaskCategoryType.AddTaskType 1100
Ektron.Services.Task.TaskCategoryType.DeleteTaskType 1100
Ektron.Services.Task.TaskCategoryType.GetTaskType 1100
Ektron.Services.Task.TaskCategoryType.UpdateTaskType 1101
Ektron.Services.Taxonomy 1351
Ektron.Services.Taxonomy.AddTaxonomyItem 1352, 1353
Ektron.Services.Taxonomy.AddTaxonomySyncFolder 1353
Ektron.Services.Taxonomy.CreateTaxonomy 1353
Ektron.Services.Taxonomy.DeleteTaxonomy 1354
Ektron.Services.Taxonomy.GetTaxonomyIdByPath 1354
Ektron.Services.Taxonomy.LoadTaxonomy 1355
Ektron.Services.Taxonomy.ReadAllAssignedCategory 1355
Ektron.Services.Taxonomy.ReadTaxonomy 1356
Ektron.Services.Taxonomy.RemoveTaxonomyFolder 1356
Ektron.Services.Taxonomy.RemoveTaxonomyItem 1357
Ektron.Services.Taxonomy.ReOrderTaxonomyItems 1357
Ektron.Services.Taxonomy.UpdateTaxonomy 1357
Ektron.Services.ThreadedDiscussion 1358
Ektron.Services.ThreadedDiscussion.AddBoard 1359
Ektron.Services.ThreadedDiscussion.AddForum 1360
Ektron.Services.ThreadedDiscussion.AddReply 1360
Ektron.Services.ThreadedDiscussion.AddTopic 1361, 1362
Ektron.Services.ThreadedDiscussion.DeleteBoard 1362
Ektron.Services.ThreadedDiscussion.DeleteForum 1363
Ektron.Services.ThreadedDiscussion.DeleteTopic 1363
Ektron.Services.ThreadedDiscussion.GetBoardCategories 1364
Ektron.Services.ThreadedDiscussion.GetBoardForums 1364
Ektron.Services.ThreadedDiscussion.GetDiscussionBoard 1365
Ektron.Services.ThreadedDiscussion.GetDiscussionBoardString 1366
Ektron.Services.ThreadedDiscussion.GetForum 1366
Ektron.Services.ThreadedDiscussion.GetRepliesForTopics 1367
Ektron.Services.ThreadedDiscussion.GetTopic 1368
Ektron.Services.ThreadedDiscussion.SearchReplies 1368
Ektron.Services.User 1369
Ektron.Services.User.ActivateUserAccount 1372
Ektron.Services.User.AddMembershipUser 1373
Ektron.Services.User.AddNewUser 1373
Ektron.Services.User.AddRolePermission 1374
Ektron.Services.User.AddUser 1374
Ektron.Services.User.AddUserGroup 1375
Ektron.Services.User.AddUserToGroup 1375
Ektron.Services.User.AutoLogInUser 1376
Ektron.Services.User.CreateBlankUserObject 1376, 1377
Ektron.Services.User.DeleteUserByID 1377
Ektron.Services.User.DeleteUserByIds 1378
Ektron.Services.User.DeleteUserFromGroup 1378
Ektron.Services.User.DeleteUserGroup 1379
Ektron.Services.User.EditUserCustomProperties 1379
Ektron.Services.User.GetActiveUser 1380
Ektron.Services.User.GetADMapping 1380
Ektron.Services.User.GetAllCmsUsers 1381
Ektron.Services.User.GetAllCustomProperties 1381
Ektron.Services.User.GetAllUserByCustomProperty 1382
Ektron.Services.User.GetAllUserGroups 1383
Ektron.Services.User.GetAllUsers 1384, 1385
Ektron.Services.User.GetGroupsUserIsIn 1385
Ektron.Services.User.GetMembershipUser 1386
Ektron.Services.User.GetUser 1386
Ektron.Services.User.GetUserbyUsername 1387
Ektron.Services.User.GetUserCustomProperties 1387
Ektron.Services.User.GetUserEmailInfo 1388

Ektron.Services.User.GetUserForFolderAdmin 1388
Ektron.Services.User.GetUserGroup 1389
Ektron.Services.User.GetUserGroupForFolderAdmin 1389
Ektron.Services.User.GetUserPermissions 1390
Ektron.Services.User.GetUserPreference 1390
Ektron.Services.User.GetUsers 1391, 1392
Ektron.Services.User.GetUsersForTask 1393
Ektron.Services.User.GetUsersNotInGroup 1393
Ektron.Services.User.IsAdmin 1394
Ektron.Services.User.IsARoleMember 1396, 1397
Ektron.Services.User.IsARoleMemberForContent 1398
Ektron.Services.User.IsARoleMemberForFolder 1398
Ektron.Services.User.LogInUser 1399
Ektron.Services.User.ResetMembershipUserPassword 1400
Ektron.Services.User.UnsubscribeUser 1401
Ektron.Services.User.UpdateCustomPropertiesItemOrder 1401
Ektron.Services.User.UpdateMapCoordinate 1402
Ektron.Services.User.UpdateUser 1403
Ektron.Services.User.UpDateUserGroup 1404
Ektron.Services.User.UpdateUserPreferences 1404
Ektron.Services.User.UserObject 1405
EktronCatalogPart class 210
about EktronCatalogPart class 210
BlogID 211
BoardID 212
CalendarDynamicParameter 212
CollectionDynamicParameter 212
ContentBlockDynamicParameter 213
DefaultCalendarID 213
DefaultCollectionID 213
DefaultContentID 213
DefaultFormID 214
EktronCatalogPart properties 211
EnableBlogControl 214
EnableCalendarControl 214
EnableCollectionControl 215
EnableContentBlockControl 215
EnableDiscussionBoardsControl 215
EnableFormBlockControl 215
EnableListSummaryControl 216
EnableThreadedDiscussionsControl 216
FolderID 216
FormBlockDynamicParameter 216
Title 217
EktronCommunityCatalogPart class 217
about EktronCommunityCatalogPart class 217
EktronCommunityCatalogPart properties 217
EnableBlogs 218
EnableDiscussionBoards 218
EnableForums 218
Title 219
EktronWebPartZone class 219
about EktronWebPartZone class 219
CmsSitePath 219
EktronWebPartZone properties 219
EkWebControl class 220
about EkWebControl class 220
Authenticated 225
CmsSitePath 226
EkWebControl 220
EkWebControl methods 220
EkWebControl properties 225
Error 226
Fill 221
GetEnumDescription 221
GetFilenameFromPath 221
GetMessage 222
GetMessage method 222
Hide 226
IsLoggedIn 227
Language 227
LanguageID 227
LocalizeString 222
loggedInUserID 227
loggedInUserName 228
LogLevel 228
RedirectFromLoginPage 223
RedirectToLoginPage 223

RegisterAllDmsMenuCssAndScripts 223
RegisterDmsMenuCSS 223
RegisterDmsMenuScript 224
RegisterJQueryScript 224
RegisterMembershipTabScript 224
RegisterThickBoxCSS 224
RegisterThickBoxScript 224
SuppressWrapperTags 228
Text 228
ToString 224
Wrap 225
WrapTag 229
EkWebPart class 229
about EkWebPart class 229
CatalogIconImageUrl 230
CreateEditorParts 230
Description 230
EkWebPart methods 230
EkWebPart properties 230
Subtitle 231
Title 231
TitleIconImageUrl 231
TitleUrl 231
WebBrowsableObject 231
EkXsltBase class 232
about EkXsltBase class 232
CustomXml 232
DisplayXslt 233
EkXsltBase 232
EkXsltBase properties 232
EnableErrorLogging 233
XmlDoc 233
XsltDoc 234
EntryAttributeData class 1485
about EntryAttributeData class 1485
AttributeTypeId 1487
CurrentValue 1487
EntryAttributeData 1485, 1486
EntryAttributeData constructor 1485
EntryAttributeData properties 1486
VersionId 1487
EntryData class 1487
about EntryData class 1487
Attributes 1490
Comment 1490
ContentStatus 1490
CurrencyId 1490
DateModified 1490
Dimensions 1490
DisableInventoryManagement 1491
EndDate 1491
EndDateAction 1491
EntryData 1488
EntryData constructor 1487
EntryData properties 1488
EntryType 1491
FolderId 1492
GoLive 1492
Html 1492
Id 1492
Image 1493
ImageThumbnail 1493
IsArchived 1493
IsBuyable 1493
IsMarkedForDeletion 1493
IsPublished 1494
IsTangible 1494
LanguageId 1494
LastEditorFirstName 1494
LastEditorId 1494
LastEditorLastName 1495
ListPrice 1495
Media 1495
Metadata 1495
NumberSold 1495
Pricing 1496
ProductType 1496
QuantityMultiple 1496

Quicklink 1496
SalePrice 1497
Sku 1497
Status 1497
StatusLanguage 1497
Summary 1497
TaxClassId 1498
TemplateId 1498
Title 1498
Url 1498
VersionId 1498
Weight 1499
EntryGroupData class 1499
about EntryGroupData class 1499
Description 1500
DisplayOrder 1500
EntryGroupData 1499, 1500
EntryGroupData constructor 1499
EntryGroupData properties 1500
EntryId 1501
GroupType 1501
Id 1501
Name 1501
EntryGroupItemData class 1501
about EntryGroupItemData class 1501
DisplayOrder 1502
EntryGroupItemData 1502
EntryGroupItemData constructor 1502
EntryGroupItemData properties 1502
EntryId 1503
GroupId 1503
Id 1503
ExchangeRateApi class 634
about ExchangeRateApi class 634
Add 635
ConvertPrice 635
Delete 636
ExchangeRateApi methods 635
GetCurrentExchangeRate 636
GetCurrentList 636
GetList 636
Update 637
ExchangeRateData class 1503
about ExchangeRateData class 1503
BaseCurrencyId 1505
ConvertPrice 1504
EffectiveDate 1505
ExchangeCurrencyId 1505
ExchangeRateData 1504
ExchangeRateData methods 1504
ExchangeRateData properties 1504
Rate 1505
Validate 1504
ExplorerDragDrop class 234
about ExplorerDragDrop class 234
AssetID 234
ContentLanguage 235
ExplorerDragDrop properties 234
FolderID 235
IsImage 235
OverrideExtension 236
TaxonomyId 236
ExtensionAuthor class 561
about ExtensionAuthor class 561
Author 562
ExtensionAuthor 561
ExtensionAuthor methods 561
ExtensionAuthor properties 562
ToString 562
ExtensionDescription class 562
about ExtensionDescription class 562
Desc 563
ExtensionDescription 563
ExtensionDescription methods 563
ExtensionDescription properties 563
ToString 563
ExtensionEvent class 563
about ExtensionEvent class 563

Content 580
ExtensionEvent methods 564
ExtensionEvent properties 580
Folder 581
Form 581
GetActiveDirectoryUserProxy 566
GetAssetProxy 566
GetBlogProxy 566
GetBusinessRulesProxy 567
GetCalendarEventTypeProxy 567
GetCalendarProxy 567
GetCommunityGroupProxy 567
GetContentProxy 567
GetContentRatingProxy 568
GetCustomFieldsProxy 568
GetFavoritesProxy 568
GetFlagProxy 568
GetFolderProxy 569
GetFontProxy 569
GetFormProxy 569
GetFriendsProxy 569
GetLibraryProxy 570
GetMessageBoardProxy 570
GetMetadataProxy 570
GetObjectStateAsXml 570
GetPermissionsProxy 570
GetSearchManagerProxy 571
GetSiteMapProxy 571
GetSiteProxy 571
GetTagsProxy 571
GetTaskCategoryProxy 572
GetTaskCategoryTypeProxy 572
GetTaskProxy 572
GetTaxonomyProxy 572
GetThreadedDiscussionProxy 573
GetUserProxy 573
OnAfterActionRequestSent 573
OnAfterContentAdd 573
OnAfterContentDelete 574
OnAfterContentEdit 574
OnAfterContentFlagged 574
OnAfterFolderAdd 574
OnAfterFolderDelete 574
OnAfterFolderEdit 575
OnAfterFormSubmit 575
OnAfterPublish 575
OnAfterReviewRejected 575
OnAfterUserAdd 576
OnAfterUserDelete 576
OnAfterUserEdit 576
OnAfterUserLogin 576
OnBeforeActionRequestSent 577
OnBeforeContentAdd 577
OnBeforeContentDelete 577
OnBeforeContentEdit 577
OnBeforeContentFlagged 578
OnBeforeFolderAdd 578
OnBeforeFolderDelete 578
OnBeforeFolderEdit 578
OnBeforeFormSubmit 578
OnBeforePublish 579
OnBeforeReviewRejected 579
OnBeforeUserAdd 579
OnBeforeUserDelete 579
OnBeforeUserEdit 580
OnBeforeUserLogin 580
Site 581
User 581

F

Favorites class 236, 763, 1225
about Favorites class 236, 763, 1225
AddContentFavorite 764, 1226, 1227
AddContentFavorite method 764, 1226
AddFavoriteLink 765
CacheInterval 238
DefaultUserID 239
DeleteContentFavorite 765, 1227

DeleteFavoriteLink 765
DisplayMode 239
DisplayXslt 239
DynamicParameter 240
Favorites methods 237, 763, 1226
Favorites properties 237
FavoritesTitle 240
FavoritesUriTitle 240
Fill 237
GetAllFavoriteLink 766, 767
GetAllFavoriteLink method 766
GetContentFavorites 768, 769, 1228, 1229
GetContentFavorites method 767, 1228
GetFavoriteId 769
GetFavoriteLink 769
GetFavoritesDirectory 770, 1230
IsContentFavorite 770, 1230
Link 240
LinkTarget 241
MarkupLanguage 241
MaxResults 241
MoveContentFavorite 770, 1231
MoveFavoriteLink 771
Stylesheet 242
ToString 237
UpdateFavoriteLink 771
Wrap 237
XmlDoc 242
Flag class 1231
about Flag class 1231
AddFlagObject 1232
AssignFlagToContent 1233
Flag methods 1232
GetAllFlagEntries 1233
GetContentFlagData 1234
GetFlagDataById 1234
GetFlaggedRank 1235
PurgeFlagEntries 1235
RemoveObjectObject 1235
UpdateFlagObject 1236
Flagging class 771
about Flagging class 771
AddFlagObject 772
AssignFlagToContent 772
Flagging methods 772
GetAllFlagEntries 773, 774
GetAllFlagEntries method 773
GetContentFlagData 774, 775
GetContentFlagData method 774
GetDefaultFolderFlagDef 775
GetFlagDataById 775
GetFlaggedRank 776
PurgeFlagEntries 776
RemoveObjectFlag 776
UpdateFlagObject 777
FlagInfo class 581
about FlagInfo class 581
ContentHtml 582
ContentID 582
ContentLanguage 583
ContentTitle 583
EntryId 583
ErrorMessage 583
FlagComment 583
FlagDate 583
FlagId 584
FlagInfo properties 582
FlagName 584
UserAvatar 584
UserDisplayName 584
UserFirstName 584
UserID 584
UserLastName 585
UserName 585
VisitorID 585
FlexMenu class 242
about FlexMenu class 242
AutoCollapseBranches 244

CacheInterval 244
CacheIntervalAuthenticated 245
DisplayXslt 245
EnableAjax 245
EnableMouseOverPopUp 246
EnableSmartOpen 246
EnableSubmenuLinkPathMatching 246
EnableSubmenuTemplateMatching 247
Fill 243
FlexMenu methods 243
FlexMenu properties 243
LogInfo 247
MasterControlId 247
MenuDepth 248
SetDirty 243
StartCollapsed 248
StartLevel 248
Stylesheet 249
UseAliasRealId 249
XmlDoc 249
XsltDoc 250
Folder class 683, 1050, 1236
about Folder class 683, 1050, 1236
Add 1051
AddCommunityFolder 684
AddFolder 684, 1237
CreateFolder 685, 1238
Delete 1051
DeleteFolder 1238
DeleteFolderById 685
Folder 1050
Folder methods 683, 1050, 1237
GetChildFolders 687, 688, 1239
GetChildFolders method 687, 1238
GetFolder 690, 691, 1240, 1241
GetFolder method 690, 1240
GetFolderDataWithPermission 691
GetFolderId 692
GetFolderID 1241, 1242
GetFolderId method 692
GetFolderID method 1241
GetFolderInfo 693
GetItem 1051
GetList 1051
GetPath 693, 1242
IsFolderHidden 693
RenameFolder 694, 1243
Update 1052
UpdateFolder 694, 695, 1243, 1244
UpdateFolder method 694, 1243
UpdateFolderApprovalMethod 695
FolderBreadcrumb class 250
about FolderBreadcrumb class 250
DefaultContentID 251
DefaultFolderID 251
DisplayStyle 252
DisplayXslt 252
DynamicParameter 253
Fill 250
FolderBreadcrumb methods 250
FolderBreadcrumb properties 251
LinkTarget 253
Mode 253
separator 254
XmlDoc 254
FolderInfo class 585
about FolderInfo class 585
Description 586
ErrorMessage 586
FolderInfo properties 585
ID 586
Name 586
ParentID 586
ReplicationState 586
StyleSheet 587
Type 587
FolderRequest class 1615
about FolderRequest class 1615

BreakInheritButton 1617
CategoryRequired 1618
ContentSubAssignments 1618
DomainProduction 1618
DomainStaging 1618
EnableReplication 1619
FileDirectory 1619
FlagDefIdList 1619
FlagInherited 1620
FlagInheritedFrom 1620
FolderCfldAssignments 1620
FolderDescription 1621
FolderId 1621
FolderName 1621
FolderRequest properties 1616
FolderType 1621
ImageDirectory 1622
IsDomainFolder 1622
MetaInherited 1622
MetaInheritedFrom 1623
ParentId 1623
PublishActive 1623
SiteMapPath 1624
SiteMapPathInherit 1624
StyleSheet 1624
SubscriptionProperties 1625
suppressNotification 1625
TaxonomyIdList 1625
TaxonomyInherited 1626
TaxonomyInheritedFrom 1626
TemplateFileName 1626
XmlConfiguration 1627
XmlInherited 1627
Font class 1244
about Font class 1244
AddFont 1245
DeleteFont 1245
Font methods 1245
GetAllFonts 1246
GetFont 1246
GetFontConfigList 1247
UpdateFont 1247
Form class 871, 1247
about Form class 871, 1247
AssignFormTask 871, 1248
EmailFormFieldData 872, 1249
Form methods 871, 1248
GetAllForms 873, 1250
GetAllForms method 872, 1249
GetForm 873, 1251
GetFormFieldData 874, 1252
GetFormFieldData method 873, 1251
GetFormFieldList 874, 1253
GetFormFieldQuestions 875, 1253
GetFormTitle 875, 1254
SubmitFormFieldData 875, 1254
UpdateFormFieldData 876, 1255
FormBlock class 254
about FormBlock class 254
AddValidation 257
CacheInterval 258
CopyTo 255
Count 258
Current 258
DefaultFormID 259
DynamicParameter 259
EkItem 259
Fields 260
Fill 255
FormBlock methods 254
FormBlock properties 257
FormName 260
GetEnumerator 255
GetView 255
GetViewNames 256
IncludeTags 260
IsSynchronized 261
MoveNext 256

Reset 256
StartingFolderID 261
SyncRoot 261
this 256
ToString 256
Wrap 257
FormField class 587
about FormField class 587
FieldName 588
FieldValue 588
FormField 587
FormField properties 588
FormFieldDataItem class 1627
about FormFieldDataItem class 1627
DataValue 1628
FieldName 1628
FormFieldDataItem properties 1628
FormFieldDatalist class 1628
about FormFieldDatalist class 1628
CaptionXPath 1629
DatalistItems 1629
DataNamespaces 1629
DataSelect 1630
DataSrc 1630
FormFieldDatalist properties 1629
Language 1630
Name 1631
ValueXPath 1631
FormFieldDatalistItem class 1631
about FormFieldDatalistItem class 1631
DataValue 1632
DisplayValue 1632
FormFieldDatalistItem properties 1631
IsDefault 1632
IsEnabled 1633
FormFieldDefinition class 1633
about FormFieldDefinition class 1633
Basetype 1634
ContentType 1634
DatalistName 1634
Datatype 1635
DisplayName 1635
FieldName 1635
FormFieldDefinition properties 1633
Language 1636
XPath 1636
FormFieldItem class 261
about FormFieldItem class 261
Basetype 262
ContentType 262
DatalistItems 263
Datatype 263
DisplayName 263
FieldName 263
FormFieldItem properties 262
Language 264
Value 264
FormFieldList class 1636
about FormFieldList class 1636
Datalists 1637
Fields 1637
FormFieldList properties 1637
FormInfo class 588
about FormInfo class 588
ErrorMessage 589
FormDescription 589
FormFields 589
FormInfo methods 588
FormInfo properties 589
FormTitle 589
this 588
FormRequestData class 1637
about FormRequestData class 1637
AutoFill 1638
FieldList 1639
FormRequestData properties 1638
Id 1639
IsPoll 1639

Language 1639
PostBackData 1640
PreFillFielddata 1640
TagRequired 1640
Forum class 264
about Forum class 264
AllowAnonymousPost 268
BoardID 268
CacheInterval 269
CustomOrderBy 269
DefaultObjectID 269
DynamicForumParameter 270
DynamicGroupParameter 270
DynamicObjectParameter 270
DynamicParameter 271
DynamicThreadParameter 271
EditorCSS 271
EditorToolbar 271
EnableForumQuickSelect 272
EnableThreadRating 272
ExcludeList 273
Fill 265
FilterXslt 273
Forum methods 265
Forum properties 266
GetCallbackResult 265
HideUserProfile 273
HighlightedUserRole 274
JavascriptEditorHeight 274
JavascriptEditorHTMLMode 275
JavascriptEditorToolbar 275
JavascriptEditorWidth 276
ObjectType 276
OrderBy 276
OrderDirection 277
ProfileLink 277
ProfileLinkTarget 278
RaiseCallbackEvent 265
ShowCategories 278
TaxCategories 278
TaxonomyCols 278
TaxonomyMarkupLanguage 279
Theme 279
ToString 266
Wrap 266
Friend enumeration member 556
Friends class 279, 777, 1255
about Friends class 279, 777, 1255
AcceptFriendInvite 778
AcceptPendingFriend 778, 1257
AddFriend 779, 1257, 1258
AddFriend method 778, 1257
AddPendingFriend 779, 1258
AddSelectedFriend 780, 1259
AvatarHeight 281
AvatarWidth 282
CacheInterval 282
DefaultUserID 282
DeleteFriend 780, 1259
DeletePendingFriendRequest 780, 781, 1260
DeletePendingFriendRequest method 780, 1260
DeleteSelectedFriend 781, 1261
DisplayMode 283
DisplayXslt 283
DynamicParameter 283
Fill 280
FriendLabel 284
Friends methods 280, 777, 1256
Friends properties 280
FriendsLabel 284
GetFriends 782, 783, 1262, 1263
GetFriends method 781, 1261
GetFriendsDirectory 783, 1264
GetFriendStatus 784, 1264
GetInvitedUsers 784, 1265
GetPendingFriends 784, 1266
GetSelectedFriends 785, 1266
GetSentFriendInvitations 785, 1267

InviteUsers 785, 1267
IsFriend 786, 1268
IsSelectedFriend 786, 1268
Link 284
LinkTarget 285
MarkupLanguage 285
MaxResults 285
MoveFriend 786, 1269
ShowSortingOptions 285
Stylesheet 286
ToString 280
Wrap 280
XmlDoc 286

H

How Do I... 2
HtmlEditDisplay class 286
about HtmlEditDisplay class 286
BaseJavaScript 287
BeginJavaScript 288
EndJavaScript 288
HtmlEditDisplay methods 287
MakeFieldNameFromId 288
RenderCMSFields 288
SelectedJavaScript 289
HtmlEditor class 289
about HtmlEditor class 289
Archived 293
Authenticated 294
CmsSitePath 294
ContentAction 295
ContentID 295
ContentTitle 296
ContentType 296
EditorReadyCall 297
EnableUpload 297
ExecuteContentAction 291
HtmlEditor 290
HtmlEditor methods 290
HtmlEditor properties 292
IncludeContentInfo 298
IsLoggedIn 298
Language 298
loggedInUserID 299
loggedInUserName 299
Path 300
PreferredType 300
SafeEncode 301
ShowCMSToolbar 301
ShowTitle 302
StyleSheet 302
Text 303
ToolbarLevel 303
ToolbarResetCall 304
ToString 291
TranslationFile 304
WorkareaMode 291

I

ImageControl class 305
about ImageControl class 305
DefaultImageID 306
DynamicParameter 306
Fill 305
FolderID 307
ImageControl methods 305
ImageControl properties 306
TaxonomyID 307
Title 308
Wrap 306
ImageMediaData class 1506
about ImageMediaData class 1506
ImageMediaData properties 1506
Thumbnails 1506
IndexSearch class 308
about IndexSearch class 308
AddParm 310
AddParm method 309

ButtonImgSrc 314
ButtonText 315
ContentParameter 315
CopyTo 310
Count 315
Current 316
CustomOrderBy 316
EkItems 316
EmptyResultMsg 316
EncodeContentHtml 316
Fill 310
FolderId 317
FormName 317
GetEnumerator 311
GetSearchParmDataType 311
GetSearchParmSearchType 311
GetXPathValues 311
IndexSearch 308
IndexSearch methods 309
IndexSearch properties 313
IndexSearch structures 309
IsSynchronized 317
Labelafter 318
Labelbefore 318
LabelBetween 318
LabelContains 319
LabeldateBetween 319
LabelEqual 319
LabelExactPhrase 320
LabelGreaterThan 320
Labelinputdate 320
Labelinputnumber 320
LabelLessThan 321
Labelon 321
LabelSelect 321
LinkTarget 322
MaxResults 322
MoveNext 312
OrderBy 322
OrderByDirection 323
Recursive 323
Reset 312
Search 312
Search method 312
SearchParmXML 323
ShowSearchBoxAlways 324
SyncRoot 324
this 313
Weighted 324
Wrap 313
XmlConfigId 324
XmlDoc 325
InventoryApi class 637
about InventoryApi class 637
InventoryApi methods 637
SaveInventory 637
InventoryData class 1506
about InventoryData class 1506
EntryId 1507
InventoryData properties 1507
ReorderLevel 1507
UnitsInStock 1507
UnitsOnOrder 1507
InventoryProvider class 1407
about InventoryProvider class 1407
DecreaseStockLevel 1408
GetInventory 1409
GetReorderLevel 1409
GetUnitsInStock 1410
GetUnitsOnOrder 1410
IncreaseStockLevel 1410
InventoryProvider methods 1407
SaveInventory 1411
Invite class 325
about Invite class 325
DisplayXslt 326
Fill 325
FriendMessageId 327

Invite methods 325
Invite properties 326
InviteMessageId 327
OptionalMessageText 327
Stylesheet 328
ToString 326
Wrap 326
XmlDoc 328

J

JoinGroup enumeration member 556

K

KitConfigData class 1508
about KitConfigData class 1508
BasketItemId 1509
GetGroupIndex 1508
Groups 1509
Id 1509
KitConfigData 1508
KitConfigData methods 1508
KitConfigData properties 1508
PriceModifier 1509
KitData class 1510
about KitData class 1510
KitData 1510
KitData properties 1510
OptionGroups 1510

L

LanguageAPI class 328
about LanguageAPI class 328
CurrentLanguage 330
CurrentLanguageID 330
DefaultLanguage 331
DefaultLanguageID 331
Fill 329
GetLanguage 329
GetLanguageID 329
IsValid 329, 330
IsValid method 329
LangaugeIdList 331
LangaugeTitleList 331
LanguageAPI 328
LanguageAPI methods 328
LanguageAPI properties 330
MultiLanguageEnabled 331
OverrideDefault 332
SiteLanguage 332
SiteLanguageID 332
LanguageSelect class 333
about LanguageSelect class 333
Fill 333
LabelName 334
LanguageSelect methods 333
LanguageSelect properties 334
Text 334
Wrap 333
WrapTag 334
Library class 695, 1269
about Library class 695, 1269
AddLibraryItem 696
AddLoadBalanceItem 696, 1270
DeleteLibraryItem 697, 1271
DeleteLibraryItemById 697, 1271
DeleteLoadBalanceItem 697, 1272
GetAllChildLibItems 697, 1272
GetAllLoadBalancePaths 700, 1273, 1274
GetAllLoadBalancePaths method 699, 1273
GetLibraryItem 701, 1274, 1275
GetLibraryItem method 700, 1274
GetLibrarySettings 701, 1275
GetLibraryTypes 702, 1276
GetLoadBalancePath 702, 1276
InternalLibrarySearch 702, 1277
Library methods 695, 1270
UpdateLibrary 703, 1277
UpdateLibrarySettings 703, 1278

UpdateLoadBalanceSettings 703, 1278
UpdateQlinkTemplates 703, 1278
List of Developer Resources 4
ListSummary class 335
about ListSummary class 335
AddText 339
CacheInterval 340
ContentParameter 340
ContentType 340
CopyTo 335
Count 341
Current 341
Direction 341
EkItems 342
EnablePaging 342
Fill 336
FolderDescription 342
FolderID 343
FolderName 343
FolderPath 343
GetEnumerator 336
GetHtml 343
GetRssFeed 336
GetView 336
GetViewNames 337
IncludeIcons 344
IsSynchronized 344
LinkTarget 344
ListSummary methods 335
ListSummary properties 338
MarkupLanguage 345
MaxResults 345
MoveNext 337
OrderByDirection 345
OrderKey 346
Random 346
Recursive 346
Reset 337
SelTaxonomyID 347
SuppressWrapperTags 347
SyncRoot 347
this 337
ToString 338
Wrap 338
XmlDoc 347
Localization class 704
about Localization class 704
GetAddViewLanguage 705
GetAddViewLanguageForAllContent 705
GetAddViewLanguageForMenus 705
GetContentItems 706
GetFile 706
GetFiles 706
GetFilesInZipFile 707
GetFlagUrl 708
GetFlagUrl method 707
GetInterchangeFile 709
GetInterchangeFiles 709
GetInterchangeFilesInZipFile 709
GetJobs 710
GetJobs method 710
GetLocalizationUrl 711
GetTranslationUploadDirectory 711
Localization methods 704
StartExportContentForTranslation 711
StartExportFolderForTranslation 712
StartExportMenusForTranslation 712
StartImportTranslation 713
LoggedInUser enumeration member 1082
Login class 347
about Login class 347
AutoAddType 348
AutoLogin 349
Fill 348
Login methods 348
Login properties 348
OnlyAllowMemberLogin 349
Password 349

PromptLogout 349
SuppressHelpButton 350
Text 350
UserName 350

M

Map class 350
about Map class 350
Address 352
ContentId 353
CustomIcon 353
DateSearch 353
DisplayTextResult 353
DistanceUnit 354
EnableSearchResult 354
Fill 351
FolderId 354
ForceCategory 355
GeoControl 355
GetCallbackResult 351
Latitude 355
Longitude 356
Map methods 351
Map properties 351
MapCategory 356
MapProvider 356
MapStyle 356
MarkupLanguage 357
MinZoomLevel 357
PageSize 357
RaiseCallbackEvent 351
Recursive 357
SearchData 358
SelectedCategoryLogicalType 358
StartZoomLevel 358
Stylesheet 358
TemplateParamName 359
TemplateUserProfile 359
TypeControl 359
ZoomControl 359
MediaData class 1510
about MediaData class 1510
Alt 1511
FileName 1511
FilePath 1512
Height 1512
Id 1512
MediaData properties 1511
MimeType 1512
Sequence 1512
Title 1513
Width 1513
MediaGalleryData class 1513
about MediaGalleryData class 1513
Images 1514
MediaGalleryData properties 1513
ObjectId 1514
Membership class 360
about Membership class 360
AllowAvatarUpload 361
AutoLogin 362
BoardID 362
DisplayMode 362
DynamicParameter 363
Fill 360
JavascriptEditorHeight 363
JavascriptEditorWidth 363
Membership methods 360
Membership properties 360
RedirectFailedURL 364
RedirectSuccessURL 364
RegisterButtonImg 364
RegisterButtonText 364
ResetButtonImg 365
ResetButtonText 365
ResetPasswordMessageId 365
SaveButtonImg 365
SaveButtonText 366

ShowExtended 366
ShowTerms 366
TaxonomyId 366
Text 367
UserExistsMessage 367
UserID 367
UserSuccessMessage 367
UserUpdateSuccessMessage 368
Menu class 368
about Menu class 368
CacheInterval 369
DefaultMenuID 370
DisplayXslt 370
DynamicParameter 370
Fill 368
Menu methods 368
Menu properties 369
Stylesheet 371
SuppressAddEdit 371
Wrap 369
MessageBoard class 371, 787, 1279
about MessageBoard class 371, 787, 1279
AddMessageBoardEntry 788, 1280
ApproveMessage 789, 1281, 1282
ApproveMessage method 788, 1280
CacheInterval 374
CommentBoxLocation 374
CustomSpamMethod 372
DefaultObjectID 374
DeleteMessageBoardEntries 790, 791, 1283, 1284
DeleteMessageBoardEntries method 790, 1283
DeleteMessageBoardEntriesByUser 792, 793, 1285, 1286, 1287
DeleteMessageBoardEntriesByUser method 792, 1285
DeleteMessageBoardEntry 794, 795, 1288, 1289
DeleteMessageBoardEntry method 793, 1287
DisableModeration 795
DisplayXslt 375
DynamicObjectParameter 375
EnableModeration 795
EnablePaging 375
Fill 372
GetAllMessageBoardEntries 796, 797, 1290, 1291
GetAllMessageBoardEntries method 796, 1289
GetAllMessageBoardEntriesByUser 798, 799, 1293, 1294
GetAllMessageBoardEntriesByUser method 798, 1292
GetMessageBoardEntry 800, 801, 1295, 1296
GetMessageBoardEntry method 800, 1294
IsModerated 802
IsSpam 802
IsUserMessageBoardAdmin 802, 1297
MarkupLanguage 376
MaxResults 376
MessageBoard methods 372, 787, 1279
MessageBoard properties 373
Moderate 377
ObjectType 377
ProfileParamName 377
RaiseCallbackEvent 372
ShowMaxCharacter 378
SpamControlType 378
SpamTimeSpan 378
Stylesheet 379
TemplateUserProfile 379
ToString 372
UpdateMessageText 803, 804, 1298, 1299
UpdateMessageText method 803, 1297
Wrap 373
XmlDoc 379
MessageBoardInfo class 590
about MessageBoardInfo class 590
ErrorMessage 590
MessageBoardInfo 590
MessageBoardInfo properties 590
Messaging class 380, 713
about Messaging class 380, 713
DisplayXslt 381
EnablePreSearch 382
Fill 380

FriendsOnly 382
GetUnreadMessageCountForUser 713
GroupParamName 382
JavascriptEditorHeight 383
JavascriptEditorHTMLMode 383
JavascriptEditorToolbar 383
JavascriptEditorWidth 384
Messaging methods 380, 713
Messaging properties 381
ProfileParamName 384
ProfileUrl 385
RecipientParamName 385
ShowSearchOptions 385
Stylesheet 386
ToString 380
UserSearchDisplayXslt 386
Wrap 380
XmlDoc 386
Metadata class 714, 1300
about Metadata class 714, 1300
AddMetaDataType 715, 1301
AddMetaDataType method 714
DeleteMetadataType 717, 1301
GetContentMetadataList 717, 719, 1302
GetContentMetadataList method 717, 1301
GetIntegratedSearchFolder 719
GetIntegratedSearchFolders 719
GetMetadataByFolder 720
GetMetadataType 720, 1303
GetMetaDataTypes 720, 1303
Metadata methods 714, 1300
UpdateMetadataType 721, 1304
MetaData class 386
about MetaData class 386
CacheInterval 388
DefaultContentID 388
DefaultItemList 389
DynamicParameter 389
Fill 387
GenerateDublinCore 389
MetaData methods 387
MetaData properties 387
MetaDataList class 390
about MetaDataList class 390
CacheInterval 393
ContentType 394
CopyTo 391
Count 394
Current 395
EkItems 395
EnablePaging 395
ExactPhrase 395
Fill 391
FolderID 396
GetEnumerator 391
GetHtml 396
IncludeIcons 397
IsSynchronized 397
KeyWordName 397
KeyWordValue 398
KeyWordValueMatchAll 398
KeyWordValueSeparator 398
LinkTarget 399
MarkupLanguage 399
MaxNumber 399
MetaDataList methods 390
MetaDataList properties 392
MoveNext 391
OrderBy 400
Recursive 400
Reset 391
SortOrder 401
SuppressWrapperTags 401
SyncRoot 401
this 391
Wrap 392
XmlDoc 401
MicroMessage class 1070

_MicroMessageBL 1071
about MicroMessage class 1070
Add 1072
AddReply 1072
Delete 1072
GetColleagueMessageList 1073
GetColleagueMessageList method 1072
GetInstance 1076
GetItem 1073
GetList 1073
GetReplyList 1074
GetSearchList 1074
GetTimeLineMessageList 1074, 1075
GetTimeLineMessageList method 1074
GetUserMessageList 1075
IsFullTextSearchInstalled 1075
IsSpam 1075
MicroMessage 1070, 1071
MicroMessage constructor 1070
MicroMessage fields 1071
MicroMessage methods 1071
MicroMessage properties 1076
MicroMessaging class 401
about MicroMessaging class 401
CacheInterval 404
CustomSpamMethod 402
DefaultObjectID 404
DisplayXslt 405
DynamicObjectParameter 405
EnablePaging 405
Fill 402
MaxResults 406
MaxSearchResults 406
MicroMessaging methods 402
MicroMessaging properties 403
Mode 407
ProfileParamName 407
RaiseCallbackEvent 402
SpamControlType 407
SpamTimeSpan 408
Stylesheet 408
TemplateMessage 408
TemplateUserProfile 409
ToString 403
Wrap 403
XmlDoc 409
MyAccount class 409
about MyAccount class 409
CustomPropertyID 411
DefaultCountryId 411
DisplayXslt 412
EkItem 412
Fill 410
LoadingImage 412
MyAccount methods 410
MyAccount properties 410
RedirectUrl 413
Stylesheet 413
ToString 410
UseMsAjax 413
Wrap 410

N

Notification class 1060
about Notification class 1060
Notification 1060
Notification methods 1061
Send 1061
Send method 1061
NotificationAgentSetting class 1062
about NotificationAgentSetting class 1062
Add 1062
Delete 1063
GetAgent 1063
GetItem 1063
GetList 1063
GetRegisteredAgentList 1064
NotificationAgentSetting 1062

NotificationAgentSetting methods 1062
Update 1064
NotificationPreference class 1064
about NotificationPreference class 1064
Add 1065
Delete 1065
GetDefaultPreferenceList 1065
GetItem 1066
GetList 1066
GetUserNotificationListForActivity 1066
NotificationPreference 1064
NotificationPreference methods 1065
SaveUserPreferences 1066
Update 1067

O

OptionGroupData class 1514
about OptionGroupData class 1514
Image 1516
OptionGroupData 1515
OptionGroupData constructor 1514
OptionGroupData properties 1515
Options 1516
OptionGroupItemCollection class 1516
about OptionGroupItemCollection class 1516
AddRange 1517
Find 1517
FindAll 1517
FindIndex 1517
OptionGroupItemCollection 1516
OptionGroupItemCollection methods 1516
OptionGroupItemData class 1518
about OptionGroupItemData class 1518
CurrencyId 1519
ExtraText 1519
Name 1520
OptionGroupItemData 1518, 1519
OptionGroupItemData constructor 1518
OptionGroupItemData properties 1519
PriceModification 1520
PriceModificationType 1520
OrderApi class 638
about OrderApi class 638
Capture 639
Capture method 639
GetCustomerOrderList 639
GetItem 640
GetList 640, 641
GetList method 640
GetOnHoldOrderList 641
GetOnHoldOrderList method 641
GetOrderMessage 642
GetOrderMessage method 642
GetOrderPayment 643
GetOrderStatus 643
GetProductOrderList 643, 644
GetProductOrderList method 643
GetReport 644
ItemsInStock 644
MarkAsFraud 645
MarkPaymentAsSettled 645
MarkPaymentAsSettled method 645
OrderApi methods 638
ProcessOrderInventory 646
RollbackOrderInventory 646
SetOrderStatus 646
SetOrderToHold 646
SetTrackingNumber 647
OrderData class 1520
about OrderData class 1520
BillingAddressId 1521
ClientIP 1521
CouponTotal 1521
CouponUsed 1522
Currency 1522
Customer 1522
CustomerType 1522
DateCompleted 1522

DateCreated 1523
DateRequired 1523
HandlingTotal 1523
HttpHost 1523
Id 1523
OrderData properties 1520
OrderTotal 1524
Parts 1524
ShippingTotal 1524
SpecialInstructions 1524
StageId 1524
StageName 1525
Status 1525
Subtotal 1525
TaxTotal 1525
VisitorId 1525
OrderLineData class 1526
about OrderLineData class 1526
Configuration 1526
ItemType 1526
LineItemId 1527
OrderLineData properties 1526
OrderPartId 1527
PriceEach 1527
PriceSubTotal 1527
PriceTotal 1527
ProductId 1528
ProductLanguage 1528
Quantity 1528
OrderList class 414
about OrderList class 414
CacheInterval 415
DefaultUserID 416
DisplayXslt 416
DynamicOrderParameter 417
DynamicProductParameter 417
EkItem 417
Fill 414
OrderList methods 414
OrderList properties 415
PageSize 418
Stylesheet 418
TemplateProduct 418
ToString 415
Wrap 415
OrderPartData class 1528
about OrderPartData class 1528
DateShipped 1529
HandlingTotal 1529
Lines 1529
OrderId 1530
OrderPartData properties 1528
PartId 1530
PartTotal 1530
ShippingAddressId 1530
ShippingTotal 1530
Status 1531
Subtotal 1531
SupplierId 1531
TaxTotal 1531
TrackingNumber 1531

P

PackageApi class 647
about PackageApi class 647
Add 647
Delete 648
GetItem 648
GetList 648
PackageApi methods 647
Update 649
PackageData class 1532
about PackageData class 1532
Dimensions 1533
Id 1533
MaxWeight 1533
Name 1534
PackageData 1532, 1533

PackageData constructor 1532
PackageData properties 1533
PagingInfo class 1641
about PagingInfo class 1641
CurrentPage 1641
EndRow 1642
PagingInfo properties 1641
RecordsPerPage 1642
StartRow 1642
TotalPages 1642
TotalRecords 1642
PaymentAuthorization class 1534
about PaymentAuthorization class 1534
AuthorizedOn 1534
CapturedOn 1535
CreditedOn 1535
Gateway 1535
Message 1535
PaymentAuthorization properties 1534
RecurringTransactionId 1535
TransactionId 1535
VoidedOn 1536
PaymentGatewayProvider class 1412
about PaymentGatewayProvider class 1412
Amount 1415
Authorization 1416
Authorize 1412
AuthorizeAndCapture 1413
AuthorizeRecurringPayment 1413
CancelRecurringPayment 1413
CapturePreauthorization 1414
Config 1416
CurrencyCode 1416
CurrencyId 1416
GatewayData 1416
InTestMode 1416
IsSubmissionSuccess 1417
OrderId 1417
PaymentGatewayProvider methods 1412
PaymentGatewayProvider properties 1415
PaymentMethod 1417
RecurringPaymentFailed 1414
RecurringPaymentProcessed 1414
SubmissionError 1417
SupportsCheckPayments 1417
SupportsCreditCardPayments 1418
SupportsRecurringPayments 1418
VoidPreAuthorization 1414
PaymentMethod class 1536
about PaymentMethod class 1536
BillingAddress 1537
BillingFirstName 1537
BillingLastName 1537
GetPayment 1536
PaymentMethod methods 1536
PaymentMethod properties 1537
Permissions class 721, 1304
about Permissions class 721, 1304
AddItemPermission 722, 1305
DeleteItemPermission 722, 1306
DisableContentInheritance 722, 1306
DisableFolderInheritance 723, 1307
EnableContentInheritance 723, 1307
EnableFolderInheritance 723, 1308
GetUserPermissions 724, 1309
GetUserPermissions method 723, 1308
LoadPermissions 725, 1310
Permissions methods 721, 1305
UpdateItemPermission 725, 1310
PersonalizationManager class 419
about PersonalizationManager class 419
AddContentExitText 420
AddContentText 420
CurrentUserButtonText 421
Display 421
Fill 419
HeaderBackColor 421
HeaderText 422

PersonalizationManager methods 419
PersonalizationManager properties 420
PersonalizeExitText 422
PersonalizeScopeText 422
PersonalizeStartText 423
PublicUserButtonText 423
ResetLinkText 423
ResetLinkToolTip 424
UrlCompareMode 424
PhotoGallery class 425
about PhotoGallery class 425
Fill 425
PhotoGallery methods 425
PipelineStageData class 1537
about PipelineStageData class 1537
Name 1538
PipelineStageData properties 1538
Sequence 1538
StatusId 1538
Plug-in Extension API 552
Poll class 425
about Poll class 425
EnableAjax 427
Fill 426
Poll methods 426
Poll properties 427
PollID 427
WrapPoll 426
PostalCodeTaxRateData class 1539
about PostalCodeTaxRateData class 1539
PostalCode 1540
PostalCodeTaxRateData 1539, 1540
PostalCodeTaxRateData constructor 1539
PostalCodeTaxRateData properties 1540
RegionId 1540
PostHistory class 428
about PostHistory class 428
BoardID 429
CacheInterval 429
Fill 428
MaxNumber 430
PostHistory methods 428
PostHistory properties 429
ToString 428
URLPath 430
UserID 430
Wrap 429
PostLoginEventArgs class 431
about PostLoginEventArgs class 431
PricingData class 1540
about PricingData class 1540
CurrencyPricelist 1545
GetCurrencyById 1542
GetCurrencyByIso 1542
GetCurrencyByIsoNumber 1543
GetListPrice 1543
GetListPriceByCurrencyId 1543
GetListPriceByIsoNumber 1543
GetSalePrice 1543
GetSalePriceByCurrencyId 1544
GetSalePriceByIsoNumber 1544
IsRecurringPrice 1545
PricingData 1541
PricingData constructor 1541
PricingData methods 1542
PricingData properties 1544
Recurrence 1545
Validate 1544
Product class 431
about Product class 431
CacheInterval 433
DefaultProductID 434
DisplayXslt 434
DynamicParameter 434
EkItems 435
Fill 432
ImageGallery 435
OverrideXslt 436

Product methods 431
Product properties 432
RedirectToBasketPage 432
ShowAddToCart 436
Stylesheet 436
TemplateCart 437
ToString 432
Wrap 432
ProductCouponEntryData class 1574
about ProductCouponEntryData class 1574
EntryType 1575
ProductCouponEntryData 1575
ProductCouponEntryData constructor 1574
ProductCouponEntryData properties 1575
ProductData class 1545
about ProductData class 1545
ProductData 1546
ProductData constructor 1545
ProductData properties 1546
Variants 1546
ProductList class 437
about ProductList class 437
DisplayXslt 439
DynamicParameter 440
EkItems 440
Fill 438
IdList 440
LoadingImage 441
PageSize 441
PageSpan 441
ProductCriteria 442
ProductList methods 437
ProductList properties 438
Recursive 442
SortMode 442
SourceId 443
SourceType 443
Stylesheet 444
TaxonomyDepth 444
TemplateProduct 445
ToString 438
UseMsAjax 445
Wrap 438
ProductSearch class 445
about ProductSearch class 445
ButtonImgSrc 449
ButtonText 450
CacheInterval 450
CatalogId 450
CustomOrderBy 451
DisplayXslt 451
DynamicProductParameter 451
EnableAdvancedLink 452
EndCount 452
Fill 446
GetCallbackResult 446
IsInWorkArea 452
LinkTarget 453
LoadingImage 453
MaxCharacters 453
MaxTeaserLength 454
OrderBy 454
OrderDirection 455
ProductSearch 446
ProductSearch methods 446
ProductSearch properties 447
RaiseCallbackEvent 447
Recursive 455
ResultCount 455
ResultDataTable 456
ResultsPageSize 456
ResultTagId 456
ResultXml 457
SearchText 457
ShowCategories 457
ShowCustomSummary 458
ShowSearchBoxAlways 458
StartCount 459

Stylesheet 459
TaxCategories 459
TaxonomyOperator 459
TemplateCart 460
TemplateProduct 460
TextBoxSize 461
TimeTaken 461
Wrap 447
XmlDoc 461
ProductTypeApi class 649
about ProductTypeApi class 649
Add 650
AddProductAttribute 650
Delete 650
DeleteProductAttribute 650, 651
DeleteProductAttribute method 650
GetFolderProductTypeList 651
GetItem 651, 652
GetItem method 651
GetProductAttribute 652
GetProductAttributeList 652
ProductTypeApi methods 649
Update 652
UpdateProductAttribute 652
ProductTypeAttributeData class 1546
about ProductTypeAttributeData class 1546
ActiveStatus 1548
DataType 1548
DefaultValue 1548
DisplayOrder 1548
Id 1549
Name 1549
ProductTypeAttributeData 1547
ProductTypeAttributeData constructor 1547
ProductTypeAttributeData properties 1548
ProductTypeData class 1549
about ProductTypeData class 1549
Attributes 1550
DefaultThumbnails 1550
EntryClass 1550
ProductTypeData properties 1549
Providers API 1406

R

Recommendation class 462
about Recommendation class 462
CacheInterval 463
DefaultProductID 464
DisplayXslt 464
DynamicCartParameter 465
DynamicProductParameter 465
EkItem 466
Fill 462
PageSize 466
Recommendation methods 462
Recommendation properties 463
RecommendationType 466
Stylesheet 467
TemplateCart 467
TemplateProduct 467
ToString 462
Wrap 462
RecommendationApi class 653
about RecommendationApi class 653
Add 653
Delete 653
DeleteRecommendationByEntry 654
GetItem 654
RecommendationApi methods 653
Update 654
UpdateCrossSell 654
UpdateUpSell 655
RecommendationItemData class 1550
about RecommendationItemData class 1550
EntryLanguage 1552
EntryType 1552
Html 1552
Image 1553

ImageThumbnail 1553
Pricing 1553
Quicklink 1553
RecommendationItemData 1551
RecommendationItemData constructor 1551
RecommendationItemData properties 1552
Summary 1553
Title 1553
Type 1554
RegionApi class 655
about RegionApi class 655
Add 655
CanDelete 656
Delete 656
GetItem 656
GetList 657
RegionApi methods 655
Update 657
RegionData class 1554
about RegionData class 1554
Code 1556
CountryId 1556
Enabled 1556
Id 1557
Name 1557
RegionData 1555
RegionData constructor 1554
RegionData methods 1555
RegionData properties 1556
ToString 1555
Validate 1555
RegionTaxRateData class 1557
about RegionTaxRateData class 1557
RegionData 1558
RegionTaxRateData 1558
RegionTaxRateData constructor 1557
RegionTaxRateData properties 1558
ReportAnalytics class 1006
about ReportAnalytics class 1006
GetAbsoluteUniqueVisitorsTrend 1008
GetAbsoluteUniqueVisitorsTrend method 1007
GetAdVersions 1009
GetAdVersions method 1009
GetAveragePageViewsTrend 1010
GetAveragePageViewsTrend method 1010
GetBounceRateTrend 1011
GetBounceRateTrend method 1011
GetBrowsers 1012
GetBrowsers method 1012
GetCampaigns 1013
GetCampaigns method 1013
GetCmsSearchTerms 1014
GetConnectionSpeeds 1014, 1015
GetConnectionSpeeds method 1014
GetContentbyTitle 1015, 1016
GetContentbyTitle method 1015
GetContentDetail 1017
GetContentDetail method 1016
GetFlashVersions 1018
GetFlashVersions method 1017
GetHostnames 1019
GetHostnames method 1018
GetJavaSupport 1020
GetJavaSupport method 1020
GetKeywords 1021
GetKeywords method 1021
GetLanguages 1022
GetLanguages method 1022
GetLocations 1023
GetLocations method 1023
GetNetworkLocations 1024
GetNetworkLocations method 1024
GetNewVsReturningVisitors 1025
GetNewVsReturningVisitors method 1025
GetOperatingSystems 1026, 1027
GetOperatingSystems method 1026
GetPageViewsTrend 1027, 1028
GetPageViewsTrend method 1027

GetPlatforms 1028, 1029
GetPlatforms method 1028
GetScreenColors 1029, 1030
GetScreenColors method 1029
GetScreenResolutions 1030, 1031
GetScreenResolutions method 1030
GetTimeOnSiteTrend 1031, 1032
GetTimeOnSiteTrend method 1031
GetTopContent 1032, 1033
GetTopContent method 1032
GetTopExitPages 1034
GetTopExitPages method 1033
GetTopLandingPages 1035
GetTopLandingPages method 1034
GetVisitsTrend 1036
GetVisitsTrend method 1035
ReportAnalytics 1006
ReportAnalytics methods 1006
RequestInfo class 591
about RequestInfo class 591
ReviewInfo class 591
about ReviewInfo class 591
AverageRating 592
ContentHtml 592
ContentID 592
ContentLanguage 592
ContentTitle 592
ErrorMessage 593
ID 593
NumberofRatings 593
Rating 593
RatingDate 593
ReviewInfo properties 591
State 593
UserAvatar 594
UserComments 594
UserDisplayName 594
UserFirstName 594
UserID 594
UserLastName 594
UserName 595
VisitorID 595
RssAggregator class 468
about RssAggregator class 468
CopyTo 469
Count 471
Current 471
EkItems 471
Fill 469
GetEnumerator 469
GetView 469
GetViewNames 469
IsSynchronized 471
LinkTarget 472
MaxResults 472
MoveNext 470
Reset 470
RssAggregator methods 468
RssAggregator properties 470
SyncRoot 472
this 470
URL 473
WrapTag 473
RssItemBase class 473
about RssItemBase class 473
DatePosted 474
Description 474
Link 474
RssItemBase fields 473
Title 474

S

SchedulerStyles class 551
about SchedulerStyles class 551
Search class 474
about Search class 474
ButtonImgSrc 480
ButtonText 480

ContentFieldsetLegend 480
CopyTo 476
Count 481
Current 481
Display 481
EkItems 481
EmptyResultMsg 482
EnableAdvancedLink 482
EnableAjax 482
EnableArchived 483
EnableArchivedBtn 483
EnableContent 483
EnableContentBtn 483
EnableContentFieldset 484
EnableDMSAssets 484
EnableDMSAssetsBtn 484
EnableFolderFieldset 485
EnableForms 485
EnableFormsBtn 485
EnableLibFiles 486
EnableLibFilesBtn 486
EnableLibHyperlinks 486
EnableLibHyperlinksBtn 486
EnableLibImages 487
EnableLibImagesBtn 487
EnableLibraryFieldset 487
Fill 476
FolderFieldsetLegend 488
FolderID 488
FolderPath 488
GetEnumerator 476
IncludeIcons 489
IsSynchronized 489
LibraryFieldsetLegend 489
LinkTarget 490
LoadSearch 476
MaxCharacters 490
MaxResults 490
MaxTeaserLength 491
MoveNext 476
OrderBy 491
OrderDirection 491
PostResultsAt 492
Recursive 492
RemoveTeaserHtml 492
Reset 477
ResultTagId 493
Search 475
Search methods 475
Search properties 477
SearchDisplayUI 477
SearchText 493
SearchType 493
ShowDate 493
ShowExtendedSearch 494
ShowSearchBoxAlways 494
ShowSearchOptions 494
Stylesheet 495
SyncRoot 495
TargetFolderDescriptionList 495
TextBoxSize 496
this 477
Wrap 477
XmlDoc 496
SearchAnalytics class 1036
about SearchAnalytics class 1036
SearchAnalytics 1037
SearchManager class 899, 1311
about SearchManager class 899, 1311
Execute 900, 901
Execute method 900
ExecuteSearch 1312
ExecuteSearch method 1312
Search 903, 904, 905, 906, 1313, 1314
Search method 902, 1313
SearchForCategories 906, 907, 1315
SearchForCategories method 906
SearchManager methods 900, 1311

SearchTable 908, 1315
SearchTable method 907
Server Controls API 7
ShipmentProvider class 1419
about ShipmentProvider class 1419
AccountNumber 1421
GetRates 1419
GetServiceTypes 1420
GetTrackingUrl 1420
IsTrackingSupported 1421
Key 1421
MeterNumber 1421
Password 1421
ServiceUrl 1422
ShipmentProvider methods 1419
ShipmentProvider properties 1420
TransactionId 1422
ShippingBaseData class 1559
about ShippingBaseData class 1559
Id 1559
Name 1560
ProviderService 1560
ShippingBaseData 1559
ShippingBaseData properties 1559
ShippingMethodApi class 657
about ShippingMethodApi class 657
Add 658
ShippingMethodApi methods 657
Update 658
ShippingMethodData class 1580
about ShippingMethodData class 1580
DisplayOrder 1580
IsActive 1581
ShippingMethodData 1580
ShippingMethodData properties 1580
ShippingOptionData class 1560
about ShippingOptionData class 1560
CurrencyId 1561
ShippingFee 1561
ShippingOptionData 1560
ShippingOptionData properties 1560
Site class 725, 1316
about Site class 725, 1316
BreakTaskPermissionInherit 726, 1317
DeleteCookie 727
GetAllActiveLanguages 727, 1317
GetAllLanguages 727, 728, 1318
GetAllLanguages method 727, 1318
GetClientDayNames 728, 1319
GetClientMonthNames 728, 1319
GetCookieValue 728, 1320
GetLanguage 729, 1320
GetLanguages 729, 1321
GetSiteVariables 729, 1321
GetTaskPermission 730, 1322
IsTaskPermissionInherited 730, 1322
SetCookieValue 730, 1323
SetTaskPermission 731, 1323
Site methods 726, 1316
UpdateSiteEnabledLanguages 731, 1324
UpdateSiteVariables 731
SiteInfo class 595
about SiteInfo class 595
AppPath 595
IsSSL 596
Name 596
PhysicalSitePath 596
Port 596
SiteInfo properties 595
Url 596
Sitemap class 496
about Sitemap class 496
CacheInterval 497
ClassName 498
DisplayType 498
DisplayXslt 498
Fill 496
FlatTopLevel 499

FolderID 499
MaxLevel 499
Sitemap methods 496
Sitemap properties 497
StartingLevel 500
Stylesheet 500
XmlDoc 500
SiteMap class 732, 1324
about SiteMap class 732, 1324
GetFolderBreadcrumbPath 732, 1325
GetWebsiteMap 734, 1325
RenderFolderBreadcrumbHtml 735, 1326
RenderWebsiteMap 736, 1327
SiteMap methods 732, 1325
SmartMenu class 500
about SmartMenu class 500
AutoCollapseBranches 501
DisplayXslt 502
Enable508Compliance 502
Enable508NavigationHeadings 502
EnableMouseOverPopUp 503
EnableSmartOpen 503
EnableSubmenuLinkPathMatching 504
EnableSubmenuTemplateMatching 504
LaunchLinksInNewWindow 504
LogInfo 505
ShowRootFolder 505
SingleEditButton 505
SmartMenu properties 501
StartCollapsed 506
StartWithRootCollapsed 506
Stylesheet 506
SocialBar class 507
about SocialBar class 507
CacheInterval 509
DefaultObjectID 509
DisplayXslt 510
DynamicObjectParameter 510
EkItems 510
Fill 507
GetCallbackResult 508
IncludeIcons 511
InviteURL 511
Items 511
MarkupLanguage 512
MessagingURL 512
ObjectType 512
SocialBar methods 507
SocialBar properties 508
Stylesheet 512
SuppressWrapperTags 513
ToString 508
Wrap 508
XmlDoc 513
Subscription class 736
about Subscription class 736
AddEmailFrom 737
AddSubscription 738
AddSubscriptionMessage 738
DeleteEmailFrom 738
DeleteSubscription 738
DeleteSubscriptionMessage 739
GetAllActiveSubscriptions 739
GetAllEmailFrom 739
GetAllSubscriptionMessages 739
GetAllSubscriptions 740
GetFolderInheritedFrom 740
GetSubscription 740
GetSubscriptionMessage 740
GetSubscriptionMessagesForType 741
GetSubscriptionPropertiesForContent 741
GetSubscriptionPropertiesForFolder 741
GetSubscriptionsForContent 742
GetSubscriptionsForFolder 742
SendWebAlert 742
Subscription methods 737
UpdateEmailFrom 743
UpdateSubscription 743

UpdateSubscriptionMessage 743
UpdateSubscriptionPropertiesForContent 743
UpdateSubscriptionPropertiesForFolder 744

T

TagCloud class 513
about TagCloud class 513
CacheInterval 515
DisplayXslt 516
DynamicParameter 516
Fill 514
MarkupLanguage 516
MaxTagsDisplayed 517
OrderBy 517
OrderByDirection 517
Stylesheet 518
TagCloud methods 514
TagCloud properties 514
TagTemplate 518
TagTemplateTarget 519
TagType 519
ToString 514
Wrap 514
XmlDoc 519
Tags class 805, 1327
about Tags class 805, 1327
AddTag 806, 1329
AddTagToCommunityGroup 807, 1330, 1331
AddTagToCommunityGroup method 806, 1329
AddTagToContent 808, 809, 1332, 1333
AddTagToContent method 808, 1332
AddTagToObject 810, 1334
AddTagToObject method 809, 1333
AddTagToUser 811, 812, 1336, 1337
AddTagToUser method 811, 1335
DeleteTagById 812, 1337
DeleteTagOnObject 813, 1338
DeleteTagsByUser 813, 1338
DeleteTagsOnCommunityGroup 813, 1339
DeleteTagsOnContent 814, 1339
DeleteTagsOnObject 814, 1340
DeleteTagsOnUser 814, 1340
EditTag 814, 1341
GetAllTagRelatedContent 815, 1341
GetAllTagRelatedUsers 815, 1342
GetAllTags 816, 1343
GetAllTags method 815, 1342
GetDefaultTags 817, 1344, 1345
GetDefaultTags method 816, 1344
GetTagByID 818, 1345
GetTagByName 818, 1346
GetTagCloud 818, 1346
GetTagsForContent 818, 1347
GetTagsForObject 819, 820, 1347, 1348
GetTagsForObject method 819, 1347
GetTagsForUser 820, 821, 1349
GetTagsForUser method 820, 1348
RemoveTagAsDefault 821, 1350
SaveTagAsDefault 821, 1350
Tags methods 805, 1328
Task class 909, 1085
about Task class 909, 1085
AddTask 910, 1086
AssignTaskToUser 910, 911, 1087
AssignTaskToUser method 910, 1086
AssignTaskToUsergroup 911, 912, 1088
AssignTaskToUsergroup method 911, 1087
DeleteTask 912, 913, 1089
DeleteTask method 912, 1089
DeleteTasks 913, 1090
DeleteTasks method 913, 1090
GetTask 914, 1091
GetTasks 914, 915, 1091
GetTasks method 914
GetTasksAssignedByUser 915, 1092
GetTasksAssignedByUser method 915, 1092
GetTasksAssignedToUser 916, 917, 1093
GetTasksAssignedToUser method 916, 1093

GetTasksAssignedToUsergroup 918, 1094, 1095
GetTasksAssignedToUsergroup method 917, 1094
GetTaskState 918, 1095
SetStateForTask 919, 1095
Task methods 909, 1085
UpdateTask 919, 1096
TaskCategory class 919, 1096
about TaskCategory class 919, 1096
AddTaskCategory 920, 1097
DeleteTaskCategory 920, 1097
GetAllCategories 921, 1098
GetTaskCategory 921, 1098
TaskCategory methods 920, 1097
UpdateTaskCategory 921, 1099
TaskCategoryType class 922, 1099
about TaskCategoryType class 922, 1099
AddTaskType 922, 1100
DeleteTaskType 923, 1100
GetTaskType 923, 1100
TaskCategoryType methods 922, 1099
UpdateTaskType 923, 1101
TaxApi class 658
about TaxApi class 658
Add 659
Delete 659
GetApplicableTaxRate 659
GetApplicableTaxRateList 660
GetItem 660
GetItem method 660
GetItemByCountryId 661
GetItemByPostalCode 661
GetItemByPostalCode method 661
GetItemByPostalId 662
GetItemByRegionId 662
GetList 662
TaxApi methods 658
Update 663
TaxClassApi class 663
about TaxClassApi class 663
Add 663
Delete 664
GetItem 664
GetList 664
IsUsed 665
TaxClassApi methods 663
Update 665
TaxClassData class 1561
about TaxClassData class 1561
Id 1562
Name 1563
TaxClassData 1562
TaxClassData constructor 1561
TaxClassData properties 1562
Taxonomy class 876, 1351
about Taxonomy class 876, 1351
AddTaxonomyItem 877, 878, 1352, 1353
AddTaxonomyItem method 877, 1352
AddTaxonomySyncFolder 878, 1353
CloneTaxonomy 878
CreateTaxonomy 879, 1353
DeleteTaxonomy 879, 1354
GetTaxonomyIdByPath 879, 1354
ImportTaxonomy 880
ImportTaxonomy method 880
LoadTaxonomy 880, 1355
ReadAllAssignedCategory 881, 1355
ReadTaxonomy 881, 1356
RemoveTaxonomyFolder 881, 1356
RemoveTaxonomyItem 882, 1357
ReOrderTaxonomyItems 882, 1357
Taxonomy methods 876, 1351
UpdateTaxonomy 882, 1357
TaxonomyRequest class 1643
about TaxonomyRequest class 1643
Depth 1644
FolderId 1644
IncludeItems 1644
Preview 1645

ReadCount 1645
TaxonomyIdList 1645
TaxonomyItemType 1645
TaxonomyPath 1646
TaxonomyRequest properties 1643
TaxRateData class 1563
about TaxRateData class 1563
Id 1564
Rate 1565
TaxClassId 1565
TaxRateData 1564
TaxRateData constructor 1563
TaxRateData properties 1564
TypeId 1565
TypeItemId 1565
TaxTypeApi class 665
about TaxTypeApi class 665
Add 666
Delete 666
GetItem 666
GetList 666
TaxTypeApi methods 665
Update 667
TaxTypeData class 1565
about TaxTypeData class 1565
Id 1567
Name 1567
Precedence 1567
TaxTypeData 1566
TaxTypeData constructor 1566
TaxTypeData properties 1567
ThreadedDiscussion class 882, 1358
about ThreadedDiscussion class 882, 1358
AddBoard 884, 1359
AddForum 884, 1360
AddReply 887, 1360
AddReply method 886
AddTopic 888, 889, 1361, 1362
AddTopic method 888, 1361
DeleteBoard 889, 1362
DeleteForum 889, 1363
DeleteTopic 889, 1363
GetActiveTopics 890
GetActiveTopics method 890
GetActiveTopicsForForum 891
GetActiveUsers 891
GetBoardCategories 891, 1364
GetBoardForums 892, 1364
GetDiscussionBoard 892, 893, 1365
GetDiscussionBoard method 892, 1364
GetDiscussionBoardString 893, 1366
GetForum 893, 1366
GetRepliesForTopics 895, 1367
GetTopic 895, 1368
SearchReplies 896, 1368
SubscribeToBoard 896
SubscribeToForum 897
SubscribeToTopic 897
ThreadedDiscussion methods 883, 1358
UnSubscribeToBoard 897
UnSubscribeToForum 898
UnSubscribeToTopic 898
ThumbnailData class 1568
about ThumbnailData class 1568
Height 1568
ThumbnailData properties 1568
Width 1568
ThumbnailDefaultData class 1569
about ThumbnailDefaultData class 1569
FileNameFormat 1569
Height 1569
Id 1570
MimeType 1570
ThumbnailDefaultData properties 1569
Title 1570
Width 1570
TierPriceData class 1570
about TierPriceData class 1570

Clone 1572
CompareTo 1572
DataState 1573
Id 1573
Quantity 1573
RecurrenceId 1573
SalePrice 1574
TierPriceData 1571, 1572
TierPriceData constructor 1571
TierPriceData methods 1572
TierPriceData properties 1573
Validate 1572

U

UriFavorite class 1646
about UriFavorite class 1646
FavoriteDesc 1647
FavoriteId 1647
FavoriteName 1647
FavoriteUri 1648
UriFavorite properties 1647
UserId 1648
UrlAliasAuto class 924
about UrlAliasAuto class 924
Add 925
ClearCache 925
Delete 926
GetAutoAliasNameTypes 926
GetAutoAliasTypes 926
GetFolderItem 926
GetFolderList 927
GetFolderSource 927
GetItem 927
GetItemForContent 928
GetList 928
GetListForContent 929
GetTarget 930
GetTarget method 929
GetTaxonomyItem 930
GetTaxonomyList 931
GetTaxonomySource 931
Update 931
UrlAliasAuto methods 924
UrlAliasAutoData class 1592
about UrlAliasAutoData class 1592
AliasName 1593
AutoAliasType 1593
AutoId 1594
Example 1594
ExcludedPath 1594
FileExtension 1594
FormattedSelectedPath 1595
HostName 1595
IsEnabled 1595
LanguageId 1595
PageNameType 1595
ReplacementCharacter 1596
SelectedPath 1596
SiteId 1596
SourceId 1596
SourceName 1597
Target 1597
UrlAliasAutoData properties 1593
UrlAliasCommon class 931
about UrlAliasCommon class 931
GetAliasForContent 932
GetTarget 932, 933
GetTarget method 932
UrlAliasCommon methods 932
UrlAliasInfo class 1597
about UrlAliasInfo class 1597
AliasId 1598
CurrentAliasExtension 1598
CurrentAliasName 1598
PreviousAliasExtension 1598
PreviousAliasName 1599
UrlAliasInfo properties 1597
UrlAliasManual class 933

about UrlAliasManual class 933
Add 934
ClearCache 934
Delete 934
GetDefaultAlias 934
GetFileExtensions 935
GetItem 935
GetList 936, 937, 938, 939
GetList method 935
GetSiteList 940
GetTemplateList 940
Update 940
UrlAliasManual methods 933
ValidateName 940
UrlAliasManualData class 1599
about UrlAliasManualData class 1599
AliasId 1600
AliasName 1600
AliasPageName 1600
ContentId 1600
ContentLanguage 1601
ContentTitle 1601
DisplayAlias 1601
FileExtension 1601
HostName 1602
IsDefault 1602
IsEnabled 1602
LibraryId 1602
QueryString 1603
QueryStringAction 1603
SiteId 1603
Target 1603
UrlAliasManualData properties 1599
UrlAliasRegEx class 941
about UrlAliasRegEx class 941
Add 941
ClearCache 942
Delete 942
EvaluateExpression 942
GetItem 942
GetList 943
GetPriorityList 943
GetTarget 944
Update 944
UrlAliasRegEx methods 941
UrlAliasRegExData class 1604
about UrlAliasRegExData class 1604
Expression 1604
ExpressionMap 1605
ExpressionName 1605
IsEnabled 1605
Priority 1605
RegExId 1606
SiteId 1606
TransformedUrl 1606
UrlAliasRegExData properties 1604
UrlAliasSettings class 944
about UrlAliasSettings class 944
AutoCacheDuration 946
AutoCacheSize 946
DefaultPage 947
DisableLanguageAwareness 947
EnableAutoAliasing 947
EnableAutoCaching 947
EnabledTracing 947
EnableManualAliasing 948
EnableManualCaching 948
EnableRegExAliasing 948
EnableRegExCaching 948
IsAutoAliasEnabled 949
IsManualAliasEnabled 949
IsRegExAliasEnabled 949
ManualCacheDuration 949
ManualCacheSize 949
OverrideTemplate 950
PageExtensions 950
QueryStringAction 950
RefreshSettingsCache 945

RegExCacheDuration 950
RegExCacheSize 950
UrlAliasSettings methods 945
UrlAliasSettings properties 945
User class 958, 1076, 1369
_userService 1078
about User class 958, 1076, 1369
ActivateUserAccount 961, 1372
ActivateUserAccount method 960, 1371
Add 1078
AddMembershipUser 961, 1373
AddNewUser 963, 1373
AddRolePermission 964, 1374
AddUser 964, 1374
AddUser method 964
AddUserGroup 965, 1375
AddUserToGroup 965, 1375
AutoLogInUser 965, 1376
CreateBlankUserObject 966, 1376, 1377
CreateBlankUserObject method 966, 1376
Delete 1078
DeleteUserByID 967, 1377
DeleteUserByIds 967, 1378
DeleteUserFromGroup 967, 1378
DeleteUserGroup 967, 1379
EditUserCustomProperties 968, 1379
GetActiveUser 968, 1380
GetADMapping 968, 1380
GetAllCmsUsers 969, 1381
GetAllCustomProperties 969, 1381
GetAllUserByCustomProperty 970, 1382
GetAllUserGroups 970, 971, 1383
GetAllUserGroups method 970, 1382
GetAllUsers 971, 972, 1384, 1385
GetAllUsers method 971, 1384
GetGroupsUserIsIn 972, 1385
GetItem 1078
GetList 1079
GetMembershipUser 972, 1386
GetNonVerifyUser 973
GetNonVerifyUser method 973
GetUser 973, 1386
GetUserbyUsername 1387
GetUserByUsername 974
GetUserCustomProperties 975, 1387
GetUserEmailInfo 976, 1388
GetUserForFolderAdmin 976, 1388
GetUserGroup 976, 1389
GetUserGroupByName 976
GetUserGroupForFolderAdmin 977, 1389
GetUserPermissions 977, 1390
GetUserPreference 978, 1390
GetUsers 978, 979, 1391, 1392
GetUsers method 978, 1391
GetUsersForTask 979, 1393
GetUsersNotInGroup 980, 1393
IsAdmin 980, 1394
IsAGroupMember 980
IsARoleMember 982, 983, 1396, 1397
IsARoleMember method 981, 1394
IsARoleMemberForContent 984, 1398
IsARoleMemberForContent_FolderUserAdmin 984
IsARoleMemberForFolder 984, 1398
IsARoleMemberForFolder_FolderUserAdmin 985
IsARoleMemberForFolder_MetadataAdmin 985
LockUserAccount 986
LogInUser 986, 987, 988, 1399
LogInUser method 986
ReadCustomProperties 990
ResetMembershipUserPassword 990, 991, 1400
ResetMembershipUserPassword method 990, 1400
SetAuthenticationCookie 991
UnsubscribeUser 991, 1401
Update 1079
UpdateCustomPropertiesItemOrder 992, 1401
UpdateMapCoordinate 1402
UpdateUser 992, 1403
UpdateUser method 992, 1402

UpDateUserGroup 993, 1404
UpDateUserGroup method 993
UpdateUserPreferences 993, 1404
User 1077
User constructor 1077
User fields 1077
User methods 958, 1078, 1369
User properties 995, 1079
UserIdAuthenticated 995
UserObject 994, 1405
UserObject method 994, 1405
UserService 1079
UserInfo class 596
about UserInfo class 596
Addres 597
DisplayName 597
Editor 598
EmailAddress 598
ErrorMessage 598
FirstName 598
ID 598
Language 598
LastName 599
Latitude 599
Longitude 599
Notifications 599
Type 599
UserInfo properties 597
UserName 599
UserNotificationSetting class 1067
about UserNotificationSetting class 1067
Add 1068
Delete 1068
GetItem 1068
GetList 1069
Update 1069
UserNotificationSetting 1067
UserNotificationSetting methods 1068
VerifyValidationCode 1069
UserProfile class 520
about UserProfile class 520
AvatarHeight 521
AvatarWidth 521
CacheInterval 522
DefaultUserID 522
DisplayXslt 523
DynamicParameter 523
EnableEmailDisplay 523
Fill 520
Stylesheet 524
TagTemplate 524
TagTemplateTarget 524
TaxonomyId 524
ToString 520
UserProfile methods 520
UserProfile properties 521
Wrap 520
XmlDoc 525
Utilities class 744
about Utilities class 744
LoadPermissions 745
Utilities methods 744

W

WarehouseApi class 667
about WarehouseApi class 667
Add 668
Delete 668
GetDefaultWarehouse 668
GetItem 668
GetList 668
MarkAsDefault 669
Update 669
WarehouseApi methods 667
WarehouseData class 1575
about WarehouseData class 1575
Address 1577
Id 1577

IsDefaultWarehouse 1577
Name 1577
WarehouseData 1576
WarehouseData constructor 1576
WarehouseData properties 1576
Web Services API 1083
WebCalendar class 525, 1037
about WebCalendar class 525, 1037
Add 1039
AllowEventEditing 526
CssFile 527
DataSource 527
DefaultDisplayType 527
Delete 1039
DisplayTemplatePath 527
DisplayType 528
DynamicParameter 528
Fill 526
GetCalendar 1039
GetICalendarFeed 1039
GetPublicCalendar 1039
RequestInformationRef 528
Text 529
Update 1040
WebCalendar 1038
WebCalendar constructor 1038
WebCalendar methods 525, 1038
WebCalendar properties 526
Wrap 526
WebEvent class 1040
about WebEvent class 1040
Add 1041
CancelOccurrence 1042
CreateVariance 1042, 1043
CreateVariance method 1042
Delete 1043
GetEventOccurrenceList 1043
GetItem 1044
GetList 1044, 1045
GetList method 1044
GetNonVariantEventList 1045, 1046
GetNonVariantEventList method 1045
GetVarianceEventList 1046
GetVarianceEventList method 1046
Update 1047
WebEvent 1041
WebEvent constructor 1040
WebEvent methods 1041
WebSearch class 529
about WebSearch class 529
ButtonImgSrc 534
ButtonText 534
CustomOrderBy 534
CustomSearch 535
DisableForumSearch 535
DisplayXslt 535
DynamicContentTemplate 536
EnableAdvancedLink 536
EndCount 536
Fill 530
FolderID 536
GetCallbackResult 530
IsInWorkArea 537
LinkTarget 537
LoadSearch 530
MarkupLanguage 538
MaxCharacters 538
MaxTeaserLength 538
MultipleFolders 539
OrderBy 539
OrderDirection 539
RaiseCallbackEvent 530
Recursive 540
RemoveTeaserHtml 540
ResultCount 540
ResultDataTable 541
ResultsPageSize 541
ResultTagId 542

ResultXml 542
SearchFor 542
SearchSynonyms 543
SearchText 543
ShowCategories 544
ShowCustomSummary 544
ShowSearchBoxAlways 544
ShowSearchOptions 545
ShowSuggestedResults 545
StartCount 546
Stylesheet 546
TaxCategories 546
TaxonomyOperator 546
TextBoxSize 547
TimeTaken 547
WebSearch 529
WebSearch methods 529
WebSearch properties 531
Wrap 531
XmlDoc 547
Weight class 1577
about Weight class 1577
Amount 1579
Clone 1579
CompareTo 1579
ConvertUnits 1579
Units 1580
Weight 1578
Weight constructor 1578
Weight methods 1578
Weight properties 1579
Welcome 1
WorkSpaceGroupRequest class 1648
about WorkSpaceGroupRequest class 1648
GroupId 1649
WorkSpaceGroupRequest properties 1649
WorkSpaceUserRequest class 1649
about WorkSpaceUserRequest class 1649
UserId 1650
WorkSpaceUserRequest properties 1649