Ektron CMS400.NET API Documentation

The Server Controls API, Plug-in Extension API, .NET Assemblies and Web Services API

Copyright (c) 2006. All rights reserved.

Table of Contents

Welcome 1

List of Developer Resources 2

Ektron Namespace 5

Server Controls API **5**

Ektron.Cms Namespace 5

Ektron.Cms.Controls Namespace 5

Ektron.Cms.Controls.Personalization Namespace 7

Classes 7

EkEditorPart Class 7

Classes 8

ActiveTopics Class 9

ActiveTopics Methods 10

ActiveTopics.Fill Method 11

ActiveTopics.ToString Method 11

ActiveTopics.Wrap Method 11

ActiveTopics Properties 11

ActiveTopics.BoardID Property 12

ActiveTopics.CacheInterval Property 12

AnalyticsTracker Class 12

AnalyticsTracker Methods 13

AnalyticsTracker.Wrap Method 14

AnalyticsTracker Properties 14

AnalyticsTracker.DefaultContentID Property 14

AnalyticsTracker.DynamicParameter Property 14

AnalyticsTracker.EnableAnalytics Property 15

AssetControl Class 15

AssetControl Methods 16

AssetControl.Fill Method 16

AssetControl Properties 16

AssetControl.DynamicParameter Property 17

AssetControl.UploadType Property 17

Blog Class 17

Blog Methods 19

Blog.Fill Method 19
Blog.ToString Method 19
Blog.Wrap Method 19
Blog Properties 20
Blog.CacheInterval Property 20
Blog.PostParameter Property 20
Blog.RecentPosts Property 20
Blog.ShowHeader Property 21
BlogArchive Class 21
BlogArchive Methods 22
BlogArchive.Fill Method 22
BlogArchive.Wrap Method 23
BlogArchive Properties 23
BlogArchive.CacheInterval Property 23
BlogArchive.Text Property 23
BlogArchive.WrapTag Property 24
BlogCalendar Class 24
BlogCalendar Methods 25
BlogCalendar.Fill Method 25
BlogCalendar.Wrap Method 25
BlogCalendar Properties 26
BlogCalendar.CacheInterval Property 26
BlogCalendar.Text Property 26
BlogCalendar.WrapTag Property 26
BlogCategories Class 27
BlogCategories Methods 28
BlogCategories.Fill Method 28
BlogCategories.Wrap Method 28
BlogCategories Properties 29
BlogCategories.CacheInterval Property 29
BlogCategories.Text Property 29
BlogCategories.WrapTag Property 29
BlogEntries Class 30
BlogEntries Methods 31
BlogEntries.Fill Method 31
BlogEntries.GetRssFeed Method 32
BlogEntries.GetView Method 32
BlogEntries.GetViewNames Method 32
BlogEntries.ToString Method 32

BlogEntries.Wrap Method 33
BlogEntries Properties 33
BlogEntries.BlogPostParameter Property 33
BlogEntries.CacheInterval Property 34
BlogEntries.DisplayXslt Property 34
BlogEntries.ShowHeader Property 34
BlogEntries.WrapTag Property 34
BlogEntries.XmlDoc Property 35
BlogPost Class 35
BlogPost Methods 36
BlogPost.Wrap Method 36
BlogRecentPosts Class 36
BlogRecentPosts Methods 38
BlogRecentPosts.Fill Method 38
BlogRecentPosts.Wrap Method 38
BlogRecentPosts Properties 38
BlogRecentPosts.CacheInterval Property 39
BlogRecentPosts.NumberofPosts Property 39
BlogRecentPosts.Text Property 39
BlogRecentPosts.WrapTag Property 39
BlogRoll Class 40
BlogRoll Methods 41
BlogRoll.Fill Method 41
BlogRoll.Wrap Method 41
BlogRoll Properties 41
BlogRoll.CacheInterval Property 42
BlogRoll.Text Property 42
BlogRoll.WrapTag Property 42
BlogRSS Class 42
BlogRSS Methods 44
BlogRSS.Fill Method 44
BlogRSS Properties 44
BlogRSS.CacheInterval Property 44
BlogRSS.Text Property 44
BlogRSS.WrapTag Property 45
BreadCrumb Class 45
BreadCrumb Methods 46
BreadCrumb.Fill Method 47
BreadCrumb.GenerateTrail_HTML Method 47

ResetTrail Method 47
BreadCrumb.ResetTrail Method () 47
BreadCrumb.ResetTrail Method (ArrayList) 48
BreadCrumb.ResetTrail Method (BreadCrumbItem) 48
BreadCrumb Properties 48
BreadCrumb.DynamicParameter Property 48
BreadCrumb.MetadataName Property 49
BreadCrumbItem Class 49
BreadCrumbItem Constructor 51
BreadCrumbItem.BreadCrumbItem Constructor () 51
BreadCrumbItem.BreadCrumbItem Constructor (string, string, string, string, string) 51
BreadCrumbItem Methods 51
BreadCrumbItem.Equals Method 52
BreadCrumbItem.GetHashCode Method 52
BreadCrumbItem.ToString Method 52
BreadCrumbItem Properties 52
BreadCrumbItem.CaseSensitiveUrlComparison Property 53
BreadCrumbItem.DisplayTitle Property 53
BreadCrumbItem.IconAlt Property 53
BreadCrumbItem.IconPath Property 53
BreadCrumbItem.RawUrl Property 53
BreadCrumbItem.Target Property 54
BusinessRules Class 54
BusinessRules Methods 55
BusinessRules.Fill Method 55
BusinessRules.Wrap Method 55
BusinessRules Properties 56
BusinessRules.Parameters Property 56
BusinessRules.Result Property 56
BusinessRules.Text Property 56
Calendar Class 57
Calendar Methods 58
Calendar.Fill Method 59
Calendar.Wrap Method 59
Calendar Properties 59
Calendar.DefaultCalendarID Property 60
Calendar.DisplayType Property 60
Calendar.DynamicParameter Property 60
Calendar.EndDate Property 61

Calendar.HighlightEventID Property 61
Calendar.HighlightEventTypeID Property 61
Calendar.ShowXML Property 62
Calendar.StartDate Property 62
Calendar.Text Property 62
Collection Class 63
Collection Methods 65
Collection.CopyTo Method 66
Collection.Fill Method 66
Collection.GetEnumerator Method 66
Collection.GetRssFeed Method 66
Collection.GetView Method 66
Collection.GetViewNames Method 67
Collection.MoveNext Method 67
Collection.Reset Method 67
Collection.this Indexer 67
Collection.ToString Method 67
Collection.Wrap Method 68
Collection Properties 68
Collection.CacheInterval Property 68
Collection.ContentParameter Property 69
Collection.Count Property 69
Collection.Current Property 69
Collection.DefaultCollectionID Property 70
Collection.DynamicParameter Property 70
Collection.EkItems Property 70
Collection.EnablePaging Property 71
Collection.GetHtml Property 71
Collection.IncludeIcons Property 71
Collection.IsSynchronized Property 72
Collection.LinkTarget Property 72
Collection.MaxResults Property 72
Collection.Random Property 72
Collection.SyncRoot Property 73
Collection.XmlDoc Property 73
ContentBlock Class 73
ContentBlock Methods 75
ContentBlock.CopyTo Method 76
ContentBlock.Fill Method 76

ContentBlock.GetEnumerator Method 76
ContentBlock.MoveNext Method 76
ContentBlock.Reset Method 76
ContentBlock.this Indexer 77
ContentBlock.ToString Method 77
ContentBlock.Wrap Method 77
ContentBlock Properties 77
ContentBlock.AllowMembershipEditing Property 78
ContentBlock.CacheInterval Property 78
ContentBlock.Count Property 78
ContentBlock.Current Property 79
ContentBlock.DisplayXslt Property 79
ContentBlock.DynamicParameter Property 79
ContentBlock.EkItem Property 80
ContentBlock.IsSynchronized Property 80
ContentBlock.OverrideXslt Property 80
ContentBlock.StartingFolderID Property 80
ContentBlock.SyncRoot Property 81
ContentList Class 81
ContentList Methods 83
ContentList.CopyTo Method 84
ContentList.Fill Method 84
ContentList.GetEnumerator Method 84
ContentList.GetRssFeed Method 84
ContentList.MoveNext Method 85
ContentList.Reset Method 85
ContentList.this Indexer 85
ContentList.ToString Method 85
ContentList.Wrap Method 86
ContentList Properties 86
ContentList.CacheInterval Property 86
ContentList.ContentIds Property 87
ContentList.Count Property 87
ContentList.Current Property 87
ContentList.DefaultContentID Property 87
ContentList.Direction Property 88
ContentList.DynamicParameter Property 88
ContentList.EkItems Property 88
ContentList.EnablePaging Property 89

ContentList.GetHtml Property 89
ContentList.IncludeIcons Property 89
ContentList.IsSynchronized Property 90
ContentList.LinkTarget Property 90
ContentList.MaxResults Property 90
ContentList.MetaTag Property 90
ContentList.OrderKey Property 91
ContentList.Random Property 91
ContentList.SyncRoot Property 91
ContentList.XmlDoc Property 91
ContentRating Class 92
ContentRating Methods 94
ContentRating.Fill Method 94
ContentRating.ToString Method 94
ContentRating.Wrap Method 94
ContentRating Properties 95
ContentRating.AlreadyRatedMessage Property 95
ContentRating.BadLabel Property 96
ContentRating.CharsRemainingLabel Property 96
ContentRating.CompleteGraphDisplay Property 96
ContentRating.CookiesRequiredMessage Property 97
ContentRating.DefaultContentID Property 97
ContentRating.DynamicParameter Property 97
ContentRating.FeedbackHeader Property 98
ContentRating.GoodLabel Property 98
ContentRating.InitialGraphDisplay Property 98
ContentRating.Padding Property 99
ContentRating.RatingLevelLabel Property 99
ContentRating.ThankYouMessage Property 99
ContentRating.TotalRatingsLabel Property 100
ContentRating.UserCommentsHeader Property 100
ContentRating.ValidationMessage Property 100
DesignTimeDiagnostic Class 101
DesignTimeDiagnostic Methods 102
DesignTimeDiagnostic.Fill Method 102
DesignTimeDiagnostic Properties 102
DesignTimeDiagnostic.Authenticated Property 103
DesignTimeDiagnostic.Hide Property 103
DesignTimeDiagnostic.Language Property 103

DesignTimeDiagnostic.SuppressWrapperTags Property 103
DesignTimeDiagnostic.WrapTag Property 104
DhtmlMenu Class 104
DhtmlMenu.DhtmlMenu Constructor 106
DhtmlMenu Methods 106
DhtmlMenu.Fill Method 106
DhtmlMenu.Wrap Method 107
DhtmlMenu Properties 107
DhtmlMenu.AccessKey Property 108
DhtmlMenu.Align Property 108
DhtmlMenu.BackColor Property 108
DhtmlMenu.BorderColor Property 108
DhtmlMenu.BorderStyle Property 109
DhtmlMenu.BorderWidth Property 109
DhtmlMenu.CssClass Property 109
DhtmlMenu.DisplayXslt Property 109
DhtmlMenu.Enabled Property 109
DhtmlMenu.EnableIE6FormsFix Property 110
DhtmlMenu.Height Property 110
DhtmlMenu.HorizontalDisplay Property 110
DhtmlMenu.MenuHoverOverColor Property 110
DhtmlMenu.MenuStyle Property 111
DhtmlMenu.Stylesheet Property 111
DhtmlMenu.SubMenuBackColor Property 111
DhtmlMenu.SubMenuHoverOverColor Property 112
DhtmlMenu.SubMenuItemHeight Property 112
DhtmlMenu.SubMenuItemWidth Property 112
DhtmlMenu.SubMenuTopOffset Property 113
DhtmlMenu.TabIndex Property 113
DhtmlMenu.ToolTip Property 113
DhtmlMenu.Width Property 113
EktronCatalogPart Class 114
EktronCatalogPart Properties 114
EktronCatalogPart.DefaultContentID Property 115
EktronCommunityCatalogPart Class 115
EktronWebPartZone Class 115
EktronWebPartZone Properties 116
EktronWebPartZone.CmsSitePath Property 116
EkWebControl Class 117

EkWebControl.EkWebControl Constructor 119
EkWebControl Methods 119
EkWebControl.Fill Method 119
EkWebControl.LocalizeString Method 120
EkWebControl.RegisterClientScriptInclude Method 120
EkWebControl.RegisterStylesheet Method 120
EkWebControl.SeachHeader Method 120
EkWebControl.ToString Method 121
EkWebControl.Wrap Method 121
EkWebControl Properties 121
EkWebControl.Authenticated Property 122
EkWebControl.Error Property 122
EkWebControl.Hide Property 122
EkWebControl.IsLoggedIn Property 123
EkWebControl.Language Property 123
EkWebControl.LanguageID Property 123
EkWebControl.loggedInUserID Property 123
EkWebControl.loggedInUserName Property 124
EkWebControl.SuppressWrapperTags Property 124
EkWebControl.Text Property 124
EkWebControl.WrapTag Property 124
EkWebPart Class 125
EkXsltBase Class 125
EkXsltBase.EkXsltBase Constructor 126
EkXsltBase Methods 127
EkXsltBase.SetDirty Method 127
EkXsltBase Properties 127
EkXsltBase.DisplayXslt Property 127
EkXsltBase.XmlDoc Property 128
EkXsltBase.XsltDoc Property 128
EkXsltWebPart Class 128
FolderBreadcrumb Class 128
FolderBreadcrumb Methods 129
FolderBreadcrumb.Fill Method 130
FolderBreadcrumb Properties 130
FolderBreadcrumb.DynamicParameter Property 130
FormBlock Class 130
FormBlock Methods 132
FormBlock.CopyTo Method 133

FormBlock.Fill Method 133
FormBlock.GetEnumerator Method 133
FormBlock.GetView Method 133
FormBlock.GetViewNames Method 134
FormBlock.MoveNext Method 134
FormBlock.Reset Method 134
FormBlock.this Indexer 134
FormBlock.ToString Method 135
FormBlock.Wrap Method 135
FormBlock Properties 135
FormBlock.AddValidation Property 136
FormBlock.CacheInterval Property 136
FormBlock.Count Property 136
FormBlock.Current Property 136
FormBlock.DynamicParameter Property 137
FormBlock.EkItem Property 137
FormBlock.Fields Property 137
FormBlock.FormName Property 137
FormBlock.IncludeTags Property 138
FormBlock.IsSynchronized Property 138
FormBlock.StartingFolderID Property 138
FormBlock.SyncRoot Property 139
FormFieldItem Class 139
FormFieldItem Properties 140
FormFieldItem.Basetype Property 140
FormFieldItem.ContentType Property 140
FormFieldItem.DatalistItems Property 141
FormFieldItem.Datatype Property 141
FormFieldItem.DisplayName Property 141
FormFieldItem.FieldName Property 142
FormFieldItem.Language Property 142
FormFieldItem.Value Property 142
FormFieldItemDatalistItem Class 142
FormFieldItemDatalistItem Properties 143
FormFieldItemDatalistItem.DataValue Property 143
FormFieldItemDatalistItem.DisplayValue Property 144
Forum Class 144
Forum Methods 145
Forum.Fill Method 145

Forum.ToString Method 145
Forum.Wrap Method 146
Forum Properties 146
Forum.CacheInterval Property 146
HtmlEditDisplay Class 147
HtmlEditDisplay Methods 148
HtmlEditDisplay.BaseJavaScript Method 148
HtmlEditDisplay.BeginJavaScript Method 149
HtmlEditDisplay.EndJavaScript Method 149
HtmlEditDisplay.MakeFieldNameFromId Method 149
HtmlEditDisplay.RenderCMSFields Method 150
HtmlEditDisplay.SelectedJavaScript Method 150
HtmlEditor Class 150
HtmlEditor.HtmlEditor Constructor 154
HtmlEditor Methods 154
HtmlEditor.ExecuteContentAction Method 155
HtmlEditor.ToString Method 155
HtmlEditor.WorkareaMode Method 155
HtmlEditor Properties 156
HtmlEditor.Archived Property 157
HtmlEditor.Authenticated Property 157
HtmlEditor.CmsSitePath Property 158
HtmlEditor.ContentAction Property 158
HtmlEditor.ContentTitle Property 159
HtmlEditor.ContentType Property 159
HtmlEditor.EditorReadyCall Property 160
HtmlEditor.EnableUpload Property 160
HtmlEditor.IncludeContentInfo Property 161
HtmlEditor.IsLoggedIn Property 161
HtmlEditor.loggedInUserID Property 162
HtmlEditor.loggedInUserName Property 162
HtmlEditor.Path Property 162
HtmlEditor.PreferredType Property 163
HtmlEditor.SafeEncode Property 163
HtmlEditor.ShowCMSToolbar Property 164
HtmlEditor.ShowTitle Property 164
HtmlEditor.StyleSheet Property 165
HtmlEditor.Text Property 165
HtmlEditor.ToolbarLevel Property 166

HtmlEditor.ToolbarResetCall Property 166
HtmlEditor.TranslationFile Property 167
IndexSearch Class 167
IndexSearch.IndexSearch Constructor 171
IndexSearch Structures 171
Ektron.Cms.Controls.IndexSearch.SearchPram Structure 172
IndexSearch Methods 172
AddParm Method 173
IndexSearch.AddParm Method () 173
IndexSearch.AddParm Method (SearchPram) 173
IndexSearch.CopyTo Method 173
IndexSearch.Fill Method 173
IndexSearch.GetEnumerator Method 174
IndexSearch.GetSearchParmDataType Method 174
IndexSearch.GetSearchParmSearchType Method 174
IndexSearch.GetXPathValues Method 174
IndexSearch.MoveNext Method 175
IndexSearch.Reset Method 175
Search Method 175
IndexSearch.Search Method () 175
IndexSearch.Search Method (System.Web.HttpRequest) 175
IndexSearch.this Indexer 176
IndexSearch.Wrap Method 176
IndexSearch Properties 176
IndexSearch.ButtonImgSrc Property 177
IndexSearch.ButtonText Property 178
IndexSearch.ContentParameter Property 178
IndexSearch.Count Property 178
IndexSearch.Current Property 178
IndexSearch.EkItems Property 179
IndexSearch.EmptyResultMsg Property 179
IndexSearch.EncodeContentHtml Property 179
IndexSearch.FolderId Property 179
IndexSearch.FormName Property 180
IndexSearch.HitsMarker Property 180
IndexSearch.IsSynchronized Property 180
IndexSearch.Labelafter Property 180
IndexSearch.Labelbefore Property 181
IndexSearch.LabelBetween Property 181

IndexSearch.LabelContains Property 181
IndexSearch.LabeldateBetween Property 182
IndexSearch.LabelEqual Property 182
IndexSearch.LabelExactPhrase Property 182
IndexSearch.LabelGreaterThan Property 183
IndexSearch.Labelinputdate Property 183
IndexSearch.Labelinputnumber Property 183
IndexSearch.LabelLessThan Property 184
IndexSearch.Labelon Property 184
IndexSearch.LabelSelect Property 184
IndexSearch.LinkTarget Property 185
IndexSearch.MaxResults Property 185
IndexSearch.OrderBy Property 185
IndexSearch.OrderByDirection Property 185
IndexSearch.Recursive Property 186
IndexSearch.SearchParmXML Property 186
IndexSearch.ShowSearchBoxAlways Property 186
IndexSearch.SyncRoot Property 186
IndexSearch.Weighted Property 187
IndexSearch.XmlConfigId Property 187
IndexSearch.XmlDoc Property 187
LanguageAPI Class 187
LanguageAPI.LanguageAPI Constructor 189
LanguageAPI Methods 190
LanguageAPI.Fill Method 190
LanguageAPI.GetLanguage Method 190
LanguageAPI.GetLanguageID Method 190
IsValid Method 190
LanguageAPI.IsValid Method (int) 191
LanguageAPI.IsValid Method (string) 191
LanguageAPI Properties 191
LanguageAPI.CurrentLanguage Property 191
LanguageAPI.CurrentLanguageID Property 192
LanguageAPI.DefaultLanguage Property 192
LanguageAPI.DefaultLanguageID Property 192
LanguageAPI.LangaugeIdList Property 192
LanguageAPI.LangaugeTitleList Property 192
LanguageAPI.MultiLanguageEnabled Property 193
LanguageAPI.OverrideDefault Property 193

LanguageAPI.SiteLanguage Property 193
LanguageAPI.SiteLanguageID Property 193
LanguageSelect Class 194
LanguageSelect Methods 195
LanguageSelect.Fill Method 195
LanguageSelect.Wrap Method 195
LanguageSelect Properties 195
LanguageSelect.LabelName Property 196
LanguageSelect.Text Property 196
ListSummary Class 196
ListSummary Methods 199
ListSummary.CopyTo Method 199
ListSummary.Fill Method 199
ListSummary.GetEnumerator Method 199
ListSummary.GetRssFeed Method 200
ListSummary.GetView Method 200
ListSummary.GetViewNames Method 200
ListSummary.MoveNext Method 200
ListSummary.Reset Method 201
ListSummary.this Indexer 201
ListSummary.ToString Method 201
ListSummary.Wrap Method 201
ListSummary Properties 202
ListSummary.CacheInterval Property 202
ListSummary.ContentParameter Property 203
ListSummary.ContentType Property 203
ListSummary.Count Property 203
ListSummary.Current Property 203
ListSummary.Direction Property 204
ListSummary.EkItems Property 204
ListSummary.EnablePaging Property 204
ListSummary.FolderID Property 205
ListSummary.FolderPath Property 205
ListSummary.GetHtml Property 205
ListSummary.IncludeIcons Property 206
ListSummary.IsSynchronized Property 206
ListSummary.LinkTarget Property 206
ListSummary.MaxResults Property 207
ListSummary.OrderKey Property 207

ListSummary.Random Property 207
ListSummary.Recursive Property 208
ListSummary.SyncRoot Property 208
ListSummary.XmlDoc Property 208
Login Class 208
Login Methods 210
Login.Fill Method 210
Login Properties 210
Login.AutoAddType Property 210
Login.AutoLogin Property 211
Login.OnlyAllowMemberLogin Property 211
Login.Password Property 211
Login.PromptLogout Property 211
Login.SuppressHelpButton Property 211
Login.Text Property 212
Login.UserName Property 212
Membership Class 212
Membership Fields 214
Membership.IsPostBack Field 214
Membership Methods 214
Membership.Fill Method 215
Membership Properties 215
Membership.BoardID Property 215
Membership.DisplayMode Property 216
Membership.RedirectFailedURL Property 216
Membership.RedirectSuccessURL Property 216
Membership.RegisterButtonImg Property 216
Membership.RegisterButtonText Property 217
Membership.ResetButtonImg Property 217
Membership.ResetButtonText Property 217
Membership.ShowExtended Property 217
Membership.ShowTerms Property 218
Membership.Text Property 218
Membership.UserExistsMessage Property 218
Membership.UserID Property 218
Membership.UserSuccessMessage Property 218
Membership.UserUpdateSuccessMessage Property 219
Menu Class 219
Menu Methods 220

Menu.Fill Method 221
Menu.Wrap Method 221
Menu Properties 221
Menu.CacheInterval Property 221
Menu.DefaultMenuID Property 222
Menu.DisplayXslt Property 222
Menu.DynamicParameter Property 222
Menu.Stylesheet Property 223
MetaData Class 223
MetaData Methods 224
MetaData.Fill Method 224
MetaData Properties 224
MetaData.DefaultContentID Property 224
MetaDataList Class 225
MetaDataList Methods 227
MetaDataList.CopyTo Method 228
MetaDataList.Fill Method 228
MetaDataList.GetEnumerator Method 228
MetaDataList.MoveNext Method 228
MetaDataList.Reset Method 228
MetaDataList.this Indexer 229
MetaDataList.Wrap Method 229
MetaDataList Properties 229
MetaDataList.CacheInterval Property 230
MetaDataList.ContentType Property 230
MetaDataList.Count Property 230
MetaDataList.Current Property 230
MetaDataList.EkItems Property 231
MetaDataList.EnablePaging Property 231
MetaDataList.ExactPhrase Property 231
MetaDataList.FolderID Property 232
MetaDataList.GetHtml Property 232
MetaDataList.IncludeIcons Property 232
MetaDataList.IsSynchronized Property 233
MetaDataList.KeyWordName Property 233
MetaDataList.KeyWordValue Property 233
MetaDataList.LinkTarget Property 233
MetaDataList.MaxNumber Property 234
MetaDataList.OrderBy Property 234

MetaDataList.Recursive Property 234
MetaDataList.SortOrder Property 235
MetaDataList.SyncRoot Property 235
MetaDataList.XmlDoc Property 235
PersonalizationManager Class 235
PersonalizationManager Methods 237
PersonalizationManager.Fill Method 237
PersonalizationManager Properties 237
PersonalizationManager.AddContentExitText Property 238
PersonalizationManager.AddContentText Property 238
PersonalizationManager.CurrentUserButtonText Property 238
PersonalizationManager.Display Property 239
PersonalizationManager.HeaderBackColor Property 239
PersonalizationManager.HeaderText Property 239
PersonalizationManager.PersonalizeExitText Property 240
PersonalizationManager.PersonalizeScopeText Property 240
PersonalizationManager.PersonalizeStartText Property 240
PersonalizationManager.PublicUserButtonText Property 241
PersonalizationManager.ResetLinkText Property 241
PersonalizationManager.ResetLinkToolTip Property 242
Poll Class 242
Poll Methods 243
Poll.Fill Method 243
Poll Properties 244
Poll.EnableAjax Property 244
PostHistory Class 244
PostHistory Methods 245
PostHistory.Fill Method 246
PostHistory.ToString Method 246
PostHistory.Wrap Method 246
PostHistory Properties 246
PostHistory.BoardID Property 247
PostHistory.CacheInterval Property 247
RssAggregator Class 247
RssAggregator Methods 249
RssAggregator.CopyTo Method 249
RssAggregator.Fill Method 250
RssAggregator.GetEnumerator Method 250
RssAggregator.GetView Method 250

RssAggregator.GetViewNames Method 250
RssAggregator.MoveNext Method 251
RssAggregator.Reset Method 251
RssAggregator.this Indexer 251
RssAggregator Properties 251
RssAggregator.Count Property 251
RssAggregator.Current Property 252
RssAggregator.EkItems Property 252
RssAggregator.IsSynchronized Property 252
RssAggregator.LinkTarget Property 252
RssAggregator.MaxResults Property 253
RssAggregator.SyncRoot Property 253
RssAggregator.URL Property 253
RssItemBase Class 253
RssItemBase Fields 254
RssItemBase.DatePosted Field 254
RssItemBase.Description Field 254
RssItemBase.Link Field 255
RssItemBase.Title Field 255
Search Class 255
Search.Search Constructor 260
Search Methods 261
Search.CopyTo Method 261
Search.Fill Method 261
Search.GetEnumerator Method 261
Search.LoadSearch Method 262
Search.MoveNext Method 262
Search.Reset Method 262
Search.SearchDisplayUI Method 262
Search.this Indexer 262
Search.Wrap Method 263
Search Properties 263
Search.AllowFragments Property 265
Search.ButtonImgSrc Property 265
Search.ButtonText Property 265
Search.ContentFieldsetLegend Property 266
Search.Count Property 266
Search.Current Property 266
Search.Display Property 266

Search.EkItems Property 267
Search.EmptyResultMsg Property 267
Search.EnableAdvancedLink Property 267
Search.EnableAjax Property 267
Search.EnableArchived Property 268
Search.EnableArchivedBtn Property 268
Search.EnableContent Property 268
Search.EnableContentBtn Property 269
Search.EnableContentFieldset Property 269
Search.EnableDMSAssets Property 269
Search.EnableDMSAssetsBtn Property 269
Search.EnableFolderFieldset Property 270
Search.EnableForms Property 270
Search.EnableFormsBtn Property 270
Search.EnableLibFiles Property 271
Search.EnableLibFilesBtn Property 271
Search.EnableLibHyperlinks Property 271
Search.EnableLibHyperlinksBtn Property 271
Search.EnableLibImages Property 272
Search.EnableLibImagesBtn Property 272
Search.EnableLibraryFieldset Property 272
Search.FolderFieldsetLegend Property 273
Search.FolderID Property 273
Search.FolderPath Property 273
Search.IncludeIcons Property 274
Search.IsSynchronized Property 274
Search.LibraryFieldsetLegend Property 274
Search.LinkTarget Property 275
Search.MaxCharacters Property 275
Search.MaxResults Property 275
Search.MaxTeaserLength Property 275
Search.OrderBy Property 276
Search.OrderDirection Property 276
Search.PostResultsAt Property 276
Search.Recursive Property 277
Search.RemoveTeaserHtml Property 277
Search.ResultTagId Property 277
Search.SearchType Property 278
Search.ShowDate Property 278

Search.ShowExtendedSearch Property 278
Search.ShowSearchBoxAlways Property 278
Search.ShowSearchOptions Property 279
Search.Stylesheet Property 279
Search.SyncRoot Property 279
Search.TargetFolderDescriptionList Property 279
Search.TextBoxSize Property 280
Search.XmlDoc Property 280
Sitemap Class 280
Sitemap Methods 281
Sitemap.Fill Method 282
Sitemap Properties 282
Sitemap.CacheInterval Property 282
Sitemap.ClassName Property 282
Sitemap.DisplayType Property 283
Sitemap.FlatTopLevel Property 283
Sitemap.FolderID Property 284
Sitemap.MaxLevel Property 284
Sitemap.StartingLevel Property 284
Sitemap.XmlDoc Property 285
SmartMenu Class 285
SmartMenu Properties 286
SmartMenu.AutoCollapseBranches Property 287
SmartMenu.DisplayXslt Property 287
SmartMenu.Enable508Compliance Property 287
SmartMenu.Enable508NavigationHeadings Property 288
SmartMenu.EnableMouseOverPopUp Property 288
SmartMenu.EnableSmartOpen Property 288
SmartMenu.EnableSubmenuLinkPathMatching Property 289
SmartMenu.EnableSubmenuTemplateMatching Property 289
SmartMenu.LaunchLinksInNewWindow Property 289
SmartMenu.ShowRootFolder Property 290
SmartMenu.SingleEditButton Property 290
SmartMenu.StartCollapsed Property 290
SmartMenu.StartWithRootCollapsed Property 291
SmartMenu.Stylesheet Property 291

Plug-in Extension API **291**
Ektron.Cms.Extensibility Namespace 292
Classes 292

ContentInfo Class 293
ContentInfo Properties 294
ContentInfo.AssetID Property 295
ContentInfo.Comments Property 295
ContentInfo.ErrorMessage Property 295
ContentInfo.FileAndPath Property 295
ContentInfo.FileExtension Property 295
ContentInfo.FileName Property 296
ContentInfo.FolderID Property 296
ContentInfo.Html Property 296
ContentInfo.ID Property 296
ContentInfo.Language Property 296
ContentInfo.MimeType Property 296
ContentInfo.Status Property 297
ContentInfo.Title Property 297
ContentInfo.Type Property 297
ContentInfo.UserID Property 297
ExtensionAuthor Class 297
ExtensionDescription Class 298
ExtensionEvent Class 298
ExtensionEvent Methods 302
ExtensionEvent.GetActiveDirectoryUserProxy Method 304
ExtensionEvent.GetAssetProxy Method 304
ExtensionEvent.GetBlogProxy Method 304
ExtensionEvent.GetBusinessRulesProxy Method 305
ExtensionEvent.GetCalendarEventTypeProxy Method 305
ExtensionEvent.GetCalendarProxy Method 305
ExtensionEvent.GetContentProxy Method 305
ExtensionEvent.GetContentRatingProxy Method 306
ExtensionEvent.GetCustomFieldsProxy Method 306
ExtensionEvent.GetFolderProxy Method 306
ExtensionEvent.GetFontProxy Method 306
ExtensionEvent.GetFormProxy Method 306
ExtensionEvent.GetLibraryProxy Method 307
ExtensionEvent.GetMetadataProxy Method 307
ExtensionEvent.GetObjectStateAsXml Method 307
ExtensionEvent.GetPermissionsProxy Method 307
ExtensionEvent.GetSearchManagerProxy Method 308
ExtensionEvent.GetSiteMapProxy Method 308

ExtensionEvent.GetSiteProxy Method 308
ExtensionEvent.GetTaskCategoryProxy Method 308
ExtensionEvent.GetTaskCategoryTypeProxy Method 308
ExtensionEvent.GetTaskProxy Method 309
ExtensionEvent.GetThreadedDiscussionProxy Method 309
ExtensionEvent.GetUserProxy Method 309
ExtensionEvent.OnAfterContentAdd Method 309
ExtensionEvent.OnAfterContentDelete Method 310
ExtensionEvent.OnAfterContentEdit Method 310
ExtensionEvent.OnAfterFolderAdd Method 310
ExtensionEvent.OnAfterFolderDelete Method 310
ExtensionEvent.OnAfterFolderEdit Method 311
ExtensionEvent.OnAfterFormSubmit Method 311
ExtensionEvent.OnAfterPublish Method 311
ExtensionEvent.OnAfterUserAdd Method 311
ExtensionEvent.OnAfterUserDelete Method 312
ExtensionEvent.OnAfterUserEdit Method 312
ExtensionEvent.OnAfterUserLogin Method 312
ExtensionEvent.OnBeforeContentAdd Method 312
ExtensionEvent.OnBeforeContentDelete Method 312
ExtensionEvent.OnBeforeContentEdit Method 313
ExtensionEvent.OnBeforeFolderAdd Method 313
ExtensionEvent.OnBeforeFolderDelete Method 313
ExtensionEvent.OnBeforeFolderEdit Method 313
ExtensionEvent.OnBeforeFormSubmit Method 314
ExtensionEvent.OnBeforePublish Method 314
ExtensionEvent.OnBeforeUserAdd Method 314
ExtensionEvent.OnBeforeUserDelete Method 314
ExtensionEvent.OnBeforeUserEdit Method 315
ExtensionEvent.OnBeforeUserLogin Method 315
ExtensionEvent Properties 315
ExtensionEvent.Content Property 315
ExtensionEvent.Folder Property 316
ExtensionEvent.Form Property 316
ExtensionEvent.Site Property 316
ExtensionEvent.User Property 316
FolderInfo Class 316
FolderInfo Properties 317
FolderInfo.Description Property 318

FolderInfo.ErrorMessage Property 318
FolderInfo.ID Property 318
FolderInfo.Name Property 318
FolderInfo.ParentID Property 318
FolderInfo.ReplicationState Property 319
FolderInfo.StyleSheet Property 319
FolderInfo.Type Property 319
FormField Class 319
FormField Properties 320
FormField.FieldName Property 320
FormField.FieldValue Property 320
FormInfo Class 320
FormInfo Methods 321
FormInfo.this Indexer 322
FormInfo Properties 322
FormInfo.ErrorMessage Property 322
FormInfo.FormDescription Property 322
FormInfo.FormFields Property 322
FormInfo.FormTitle Property 323
SiteInfo Class 323
SiteInfo Properties 324
SiteInfo.AppPath Property 324
SiteInfo.IsSSL Property 324
SiteInfo.Name Property 324
SiteInfo.PhysicalSitePath Property 324
SiteInfo.Port Property 325
SiteInfo.Url Property 325
UserInfo Class 325
UserInfo Properties 326
UserInfo.DisplayName Property 326
UserInfo.Editor Property 327
UserInfo.EmailAddress Property 327
UserInfo.ErrorMessage Property 327
UserInfo.FirstName Property 327
UserInfo.ID Property 327
UserInfo.Language Property 327
UserInfo.LastName Property 328
UserInfo.Notifications Property 328
UserInfo.Type Property 328

UserInfo.UserName Property 328
WebServiceProxy Class 328
.NET Assemblies 329
API Namespace 329
API.Analytics Namespace 330
Classes 330
Analytics Class 330
Analytics Methods 332
Analytics.GetSessionRank Method 332
Analytics.HardDumpAnalyticsBuffer Method 332
Analytics.QueryAnalytics Method 332
Analytics.SoftDumpAnalyticsBuffer Method 333
TrackVisit Method 333
Analytics.TrackVisit Method (Integer, Integer, String, Integer, String, String) 333
Analytics.TrackVisit Method (System.Web.UI.Page, Integer) 334
Analytics Properties 334
Analytics.AnalyticsEnabled Property 334
Analytics.AppendToIISLog Property 335
Analytics.InMemoryContentHitsData Property 335
Analytics.RecordsBeforeWrite Property 335
Analytics.TimeBeforeWrite Property 335
API.Calendar Namespace 336
Classes 336
Calendar Class 336
CalendarEvent Class 337
CalendarEventType Class 337
API.Content Namespace 337
Classes 338
Asset Class 338
Asset Methods 339
Asset.GetAssetMgtConfigInfo Method 340
Asset.GetAssetMgtInfo Method 340
Asset.GetAssetSupertypes Method 340
Asset.GetCreateDragDropSnippet Method 341
Asset.GetCreateEktronExplorerSnippet Method 341
Asset.GetCreateSnippet Method 341
Asset.GetDragDropWOSnippet Method 341
Asset.GetEditSnippet Method 342
Asset.GetMyAssetMgtID Method 342

Asset.GetViewUrl Method 342
Asset.MakeCopyDMSAsset Method 343
Asset.SetAssetMgtConfigInfo Method 343
Blog Class 344
Blog Methods 345
Blog.AddBlog Method 346
Blog.AddCommentForPost Method 346
Blog.AddPost Method 346
Blog.DeleteBlog Method 347
Blog.DeletePost Method 347
Blog.GetBlankBlogPostData Method 347
GetBlog Method 348
Blog.GetBlog Method (Integer, String, Integer, Integer) 348
Blog.GetBlog Method (String, String, Integer, Integer) 349
GetBlogData Method 349
Blog.GetBlogData Method (FolderData) 350
Blog.GetBlogData Method (Integer) 350
Blog.GetBlogData Method (Integer, DateTime) 351
Blog.GetBlogData Method (Integer, DateTime, Integer, Integer) 351
Blog.GetBlogDataForPost Method 352
Blog.GetBlogPostData Method 352
GetBlogRoll Method 352
Blog.GetBlogRoll Method (Integer) 353
Blog.GetBlogRoll Method (String) 353
Blog.GetPost Method 354
Content Class 354
Content Methods 358
AddContent Method 360
Content.AddContent Method (String, String, String, String, String, String, Integer, Object, Object, Object) 360
Content.AddContent Method (String, String, String, String, String, String, Integer, Object, Object, Object, Integer) 361
Content.AddContent Method (String, String, String, String, String, String, Integer, Object, Object, Object, Integer, Integer) 361
AddTaxonomyItem Method 362
Content.AddTaxonomyItem Method (TaxonomyContentRequest) 362
Content.AddTaxonomyItem Method (TaxonomyRequest) 362
Content.AddTaxonomySyncFolder Method 363
Content.AddXmlConfiguration Method 363
Content.CheckOutContent Method 363

Content.CopyContentToFolder Method 364
Content.CreateTaxonomy Method 364
Content.DeleteContentItem Method 364
Content.DeleteTaxonomy Method 365
Content.DeleteXmlConfiguration Method 365
Content.GetAddViewLanguage Method 365
Content.GetAllComments Method 365
Content.GetAllTemplates Method 366
Content.GetAllUnassignedItemApprovals Method 366
Content.GetAllXmlConfigurations Method 367
Content.GetChildContent Method 367
GetContent Method 367
Content.GetContent Method (Ektron.Cms.Common.ContentRequest, System.Web.UI.Page) 368
Content.GetContent Method (Integer, ContentResultType) 368
Content.GetContent Method (Integer, Ektron.Cms.Content.EkContent.ContentResultType) 368
Content.GetContentByHistory Method 369
Content.GetContentDisplayLink Method 369
Content.GetContentForEditing Method 370
Content.GetContentReport Method 370
Content.GetContentState Method 370
Content.GetContentStatus Method 371
Content.GetDomain Method 371
Content.GetExpireContent Method 371
Content.GetTaxonomyIdByPath Method 372
Content.LoadTaxonomy Method 372
Content.MoveContentToFolder Method 372
Content.PublishContent Method 373
Content.ReadAllAssignedCategory Method 373
Content.ReadCategoryByTaxonomy Method 374
Content.ReadTaxonomy Method 374
Content.RelicateXmlConfiguration Method 374
Content.RemoveTaxonomyFolder Method 375
Content.RemoveTaxonomyItem Method 375
Content.RenameContent Method 375
Content.ReOrderTaxonomyItems Method 375
Content.RestoreHistoryContent Method 376
Content.SaveContent Method 376
Content.SubmitForDelete Method 377
Content.UndoCheckout Method 377

Content.UpdateContentMetaData Method 377
Content.UpdateTaxonomy Method 378
Content.UpdatexmlCollectionPackage Method 378
Content.UpdateXmlConfiguration Method 378
ContentRating Class 379
ContentRating Methods 380
ContentRating.GetContentRating Method 380
ContentRating.GetContentRatingAddRating Method 381
ContentRating.GetContentRatingResults Method 381
GetContentRatingStatistics Method 381
ContentRating.GetContentRatingStatistics Method (Integer, DateTime, DateTime) 382
ContentRating.GetContentRatingStatistics Method (Integer, Integer, String) 382
ContentRating.PurgeContentRatings Method 383
Form Class 383
Form Methods 384
Form.AssignFormTask Method 385
Form.EmailFormFieldData Method 385
GetAllForms Method 386
Form.GetAllForms Method () 386
Form.GetAllForms Method (Integer, String, Integer, Cms.ContentAPI.FormResultType) 386
Form.GetForm Method 387
Form.GetFormDataHistogram Method 387
GetFormFieldData Method 387
Form.GetFormFieldData Method (Integer, Integer) 387
Form.GetFormFieldData Method (Integer, String, String) 388
Form.GetFormFieldList Method 388
Form.GetFormFieldQuestions Method 389
Form.GetFormTitle Method 389
Form.SubmitFormFieldData Method 389
Form.UpdateFormFieldData Method 390
ThreadedDiscussion Class 390
ThreadedDiscussion Methods 392
ThreadedDiscussion.AddBoard Method 393
ThreadedDiscussion.AddForum Method 393
AddTopic Method 394
ThreadedDiscussion.AddTopic Method (Integer, String, String) 394
ThreadedDiscussion.AddTopic Method (Integer, String, String, Integer) 394
ThreadedDiscussion.DeleteBoard Method 395
ThreadedDiscussion.DeleteForum Method 395

ThreadedDiscussion.DeleteTopic Method 395
ThreadedDiscussion.GetBoardCategories Method 396
ThreadedDiscussion.GetBoardForums Method 396
GetDiscussionBoard Method 396
ThreadedDiscussion.GetDiscussionBoard Method (Integer) 397
ThreadedDiscussion.GetDiscussionBoard Method (String) 397
ThreadedDiscussion.GetDiscussionBoardString Method 397
ThreadedDiscussion.GetForum Method 398
ThreadedDiscussion.GetRepliesForTopics Method 398
ThreadedDiscussion.GetTopic Method 398
ThreadedDiscussion.SearchReplies Method 399
API.Search Namespace 399
Classes 399
SearchManager Class 400
SearchManager Methods 400
Execute Method 401
SearchManager.Execute Method (Ektron.Cms.ContentSearchCondition) 401
SearchManager.Execute Method (Ektron.Cms.UserSearchCondition) 402
API.User Namespace 402
Classes 402
ADUser Class 403
ADUser Methods 404
ADUser.AddADGroupToCMS Method 405
ADUser.AddADUsersToCMSByUsername Method 406
ADUser.DeSynchUserGroups Method 406
ADUser.DeSynchUsers Method 406
ADUser.GetADMapping Method 407
ADUser.GetADStatus Method 407
ADUser.GetAvailableADGroups Method 407
ADUser.GetAvailableADUsers Method 408
ADUser.GetCMSGroupsToSync Method 408
ADUser.GetCMSRelationshipsToSync Method 409
ADUser.GetCMSUsersToSync Method 409
ADUser.GetDomains Method 409
ADUser.MapCMSUserGroupToAD Method 410
ADUser.MapCMSUserToAD Method 410
ADUser.SynchCMSGroupsToAD Method 410
ADUser.SynchCMSRelationshipsToAD Method 411
ADUser.SynchCMSUsersToAD Method 411

ADUser.UpdateADMapping Method 412
ADUser.UpdateADUsersGroups Method 412
ADUser.UpdateUsersInfoFromAD Method 412
User Class 413
User Methods 416
ActivateUserAccount Method 418
User.ActivateUserAccount Method (Integer) 419
User.ActivateUserAccount Method (String, String) 419
User.AddMembershipUser Method 419
User.AddNewUser Method 420
User.AddRolePermission Method 420
AddUser Method 420
User.AddUser Method (Collection) 421
User.AddUser Method (UserData) 421
User.AddUserGroup Method 421
User.AddUserToGroup Method 421
User.AutoLogInUser Method 422
CreateBlankUserObject Method 422
User.CreateBlankUserObject Method () 422
User.CreateBlankUserObject Method (UserData) 423
User.DeleteUserByID Method 423
User.DeleteUserByIds Method 423
User.DeleteUserFromGroup Method 423
User.DeleteUserGroup Method 424
User.EditUserCustomProperties Method 424
User.GetActiveUser Method 424
User.GetADMapping Method 424
User.GetAllCmsUsers Method 425
User.GetAllCustomProperties Method 425
User.GetAllUserByCustomProperty Method 426
GetAllUserGroups Method 426
User.GetAllUserGroups Method (Ektron.Cms.Common.EkEnumeration.UserTypes, String) 426
User.GetAllUserGroups Method (String) 427
GetAllUsers Method 427
User.GetAllUsers Method (Integer, Integer) 427
User.GetAllUsers Method (UserRequestData) 428
User.GetGroupsUserIsIn Method 428
User.GetMembershipUser Method 428
User.GetUser Method 428

User.GetUserByUsername Method 429
User.GetUserCustomProperties Method 429
User.GetUserEmailInfo Method 429
User.GetUserForFolderAdmin Method 430
User.GetUserGroup Method 430
User.GetUserGroupByName Method 430
User.GetUserGroupForFolderAdmin Method 431
User.GetUserPermissions Method 431
User.GetUserPreference Method 431
GetUsers Method 432
User.GetUsers Method (Integer, String) 432
User.GetUsers Method (String, String) 433
User.GetUsersForTask Method 433
User.GetUsersNotInGroup Method 433
User.IsAdmin Method 434
IsARoleMember Method 434
User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) 435
User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean) 435
User.IsARoleMember Method (Integer) 436
User.IsARoleMember Method (Integer, Integer, Boolean) 436
User.IsARoleMemberForContent Method 436
User.IsARoleMemberForContent_FolderUserAdmin Method 437
User.IsARoleMemberForFolder Method 437
User.IsARoleMemberForFolder_FolderUserAdmin Method 438
User.IsARoleMemberForFolder_MetadataAdmin Method 438
User.LogInUser Method 438
User.ReadCustomProperties Method 439
User.ResetMembershipUserPassword Method 439
User.SetAuthenticationCookie Method 439
User.UnsubscribeUser Method 440
User.UpdateCustomPropertiesItemOrder Method 440
UpdateUser Method 441
User.UpdateUser Method (Collection) 441
User.UpdateUser Method (UserData) 441
UpDateUserGroup Method 441
User.UpDateUserGroup Method (Collection) 441
User.UpDateUserGroup Method (UserData) 442
User.UpdateUserPreferences Method 442

UserObject Method 442

User.UserObject Method (Integer) 442

User.UserObject Method (String) 443

Classes 443

BusinessRule Class 444

BusinessRule Methods 444

GetBusinessRules Method 445

BusinessRule.GetBusinessRules Method (String, String, String, String, String, String, String, Integer) 445

BusinessRule.GetBusinessRules Method (String, String, String, String, String, String, String, String) 445

CustomFields Class 446

CustomFields Methods 449

CustomFields.AddNewEditableCustomFieldAssignments Method 450

CustomFields.BoolToYesNo Method 450

CustomFields.GetEditableCustomFieldAssignments Method 450

CustomFields.GetEnhancedMetadataArea Method 451

CustomFields.GetEnhancedMetadataScript Method 451

CustomFields.GetFieldsByFolder Method 451

GetSearchProperties Method 452

CustomFields.GetSearchProperties Method (Ektron.Cms.Common.SearchAssetDispayRequest) 452

CustomFields.GetSearchProperties Method (String, String, String, String, Collection, String, String, String) 452

CustomFields.GetSearchProperties Method (System.Collections.Specialized.NameValueCollection, Ektron.Cms.Common.SearchItemCustom,), Ektron.Cms.Common.SearchItemCustom,)) 453

CustomFields.Populate_AssetRequestObjectFromForm Method 453

CustomFields.ProcessCustomFields Method 454

CustomFields.SearchAllAssets Method 454

CustomFields.WriteFilteredMetadataForEdit Method 454

CustomFields.WriteFilteredMetadataForView Method 455

CustomFields.WriteLibrarySearchExtended Method 455

CustomFields.WriteMetadataDefaultForEdit Method 455

CustomFields.WriteMetadataTypeForView Method 456

CustomFields.WritePublicSearch Method 456

CustomFields.WritePublicSearchEx Method 457

DateTimeSelector Class 458

DateTimeSelector Methods 460

DateTimeSelector.DisplayCultureDate Method 460

DateTimeSelector.DisplayCultureDateTime Method 460

DateTimeSelector.DisplayCultureTime Method 461

DateTimeSelector.RoundMinutes Method 461
DateTimeSelector Properties 461
DateTimeSelector.ClientDateSeparator Property 462
DateTimeSelector.ClientLongDatePattern Property 462
DateTimeSelector.ClientLongTimePattern Property 463
DateTimeSelector.ClientShortDatePattern Property 463
DateTimeSelector.ClientShortTimePattern Property 463
DateTimeSelector.ClientTimeSeparator Property 463
DateTimeSelector.ExtendedMeta Property 464
DateTimeSelector.FormElement Property 464
DateTimeSelector.FormName Property 464
DateTimeSelector.RequestInfo Property 464
DateTimeSelector.ServerDateSeparator Property 465
DateTimeSelector.ServerShortDatePattern Property 465
DateTimeSelector.ServerShortTimePattern Property 465
DateTimeSelector.ServerTimeSeparator Property 465
DateTimeSelector.SpanAttributes Property 465
DateTimeSelector.SpanId Property 466
DateTimeSelector.TargetDate Property 466
DateTimeSelector.ValidationType Property 466
Folder Class 466
Folder Methods 468
Folder.AddFolder Method 468
Folder.CreateFolder Method 468
Folder.DeleteFolderById Method 469
Folder.GetChildFolders Method 469
Folder.GetFolder Method 469
Folder.GetFolderDataWithPermission Method 470
GetFolderId Method 470
Folder.GetFolderId Method (Integer) 470
Folder.GetFolderId Method (String, Integer) 471
Folder.GetFolderInfo Method 471
Folder.GetPath Method 471
Folder.RenameFolder Method 472
Folder.UpdateFolder Method 472
Font Class 472
Font Methods 473
Font.AddFont Method 473
Font.DeleteFont Method 474

Font.GetAllFonts Method 474
Font.GetFont Method 474
Font.GetFontConfigList Method 474
Font.UpdateFont Method 475
Library Class 475
Library Methods 477
Library.AddLibraryItem Method 477
Library.AddLoadBalanceItem Method 478
Library.DeleteLibraryItem Method 478
Library.DeleteLibraryItemById Method 478
Library.DeleteLoadBalanceItem Method 479
Library.GetAllChildLibItems Method 479
GetAllLoadBalancePaths Method 479
Library.GetAllLoadBalancePaths Method (Integer, String) 480
Library.GetAllLoadBalancePaths Method (String) 480
GetLibraryItem Method 480
Library.GetLibraryItem Method (Integer) 481
Library.GetLibraryItem Method (Integer, Integer) 481
Library.GetLibrarySettings Method 481
Library.GetLibraryTypes Method 482
Library.GetLoadBalancePath Method 482
Library.InternalLibrarySearch Method 482
Library.UpdateLibrary Method 483
Library.UpdateLibrarySettings Method 483
Library.UpdateLoadBalanceSettings Method 483
Library.UpdateQlinkTemplates Method 484
Localization Class 484
Localization Methods 486
Localization.GetAddViewLanguage Method 487
Localization.GetAddViewLanguageForAllContent Method 487
Localization.GetAddViewLanguageForMenus Method 488
Localization.GetContentItems Method 488
Localization.GetFile Method 488
Localization.GetFiles Method 489
Localization.GetFilesInZipFile Method 489
GetFlagUrl Method 489
Localization.GetFlagUrl Method (Integer) 490
Localization.GetFlagUrl Method (LanguageData) 490
Localization.GetFlagUrl Method (LanguageDataSet.LanguageRow) 490

Localization.GetInterchangeFile Method 491
Localization.GetInterchangeFiles Method 491
Localization.GetInterchangeFilesInZipFile Method 492
GetJobs Method 492
Localization.GetJobs Method () 492
Localization.GetJobs Method (Integer) 493
Localization.GetLocalizationUrl Method 493
Localization.GetTranslationUploadDirectory Method 493
Localization.StartExportContentForTranslation Method 493
Localization.StartExportFolderForTranslation Method 494
Localization.StartExportMenusForTranslation Method 494
Localization.StartImportTranslation Method 495
Metadata Class 495
Metadata Methods 496
AddMetaDataType Method 496
Metadata.AddMetaDataType Method (Collection) 497
Metadata.AddMetaDataType Method (ContentMetaData) 497
Metadata.DeleteMetadataType Method 497
GetContentMetadataList Method 498
Metadata.GetContentMetadataList Method (Integer) 498
Metadata.GetContentMetadataList Method (Integer, Boolean) 498
Metadata.GetMetadataType Method 499
Metadata.GetMetaDataTypes Method 499
Metadata.UpdateMetadataType Method 499
Permissions Class 500
Permissions Methods 501
Permissions.AddItemPermission Method 501
Permissions.DeleteItemPermission Method 502
Permissions.DisableContentInheritance Method 502
Permissions.DisableFolderInheritance Method 502
Permissions.EnableContentInheritance Method 503
Permissions.EnableFolderInheritance Method 503
GetUserPermissions Method 503
Permissions.GetUserPermissions Method (Integer, String, Integer, String) 503
Permissions.GetUserPermissions Method (Integer, String, Integer, String, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) 504
Permissions.LoadPermissions Method 504
Permissions.UpdateItemPermission Method 505
Site Class 505

Site Methods 507
Site.BreakTaskPermissionInherit Method 507
Site.DeleteCookie Method 508
Site.GetAllActiveLanguages Method 508
Site.GetAllLanguages Method 508
Site.GetClientDayNames Method 508
Site.GetClientMonthNames Method 509
Site.GetCookieValue Method 509
Site.GetLanguage Method 509
Site.GetLanguages Method 510
Site.GetSiteVariables Method 510
Site.GetTaskPermission Method 510
Site.IsTaskPermissionInherited Method 511
Site.SetCookieValue Method 511
Site.SetTaskPermission Method 511
Site.UpdateSiteEnabledLanguages Method 512
Site.UpdateSiteVariables Method 512
SiteMap Class 512
SiteMap Methods 513
SiteMap.GetFolderBreadcrumbPath Method 513
SiteMap.GetWebsiteMap Method 514
SiteMap.RenderFolderBreadcrumbHtml Method 514
SiteMap.RenderWebsiteMap Method 514
Subscription Class 515
Subscription Methods 517
Subscription.AddEmailFrom Method 518
Subscription.AddSubscription Method 518
Subscription.AddSubscriptionMessage Method 518
Subscription.DeleteEmailFrom Method 519
Subscription.DeleteSubscription Method 519
Subscription.DeleteSubscriptionMessage Method 519
Subscription.GetAllActiveSubscriptions Method 519
Subscription.GetAllEmailFrom Method 520
Subscription.GetAllSubscriptionMessages Method 520
Subscription.GetAllSubscriptions Method 520
Subscription.GetFolderInheritedFrom Method 521
Subscription.GetSubscription Method 521
Subscription.GetSubscriptionMessage Method 521
Subscription.GetSubscriptionMessagesForType Method 521

Subscription.GetSubscriptionPropertiesForContent Method 522
Subscription.GetSubscriptionPropertiesForFolder Method 522
Subscription.GetSubscriptionsForContent Method 522
Subscription.GetSubscriptionsForFolder Method 523
Subscription.UpdateEmailFrom Method 523
Subscription.UpdateSubscription Method 523
Subscription.UpdateSubscriptionMessage Method 524
Utilities Class 524
Utilities Methods 525
Utilities.LoadPermissions Method 525

Web Services API 525
Ektron.Services Namespace 526
Ektron.Services.Task Namespace 526
Classes 527
Task Class 527
Task Methods 528
Task.AddTask Method 529
AssignTaskToUser Method 529
Task.AssignTaskToUser Method (int, int) 529
Task.AssignTaskToUser Method (int, int, int) 530
AssignTaskToUsergroup Method 530
Task.AssignTaskToUsergroup Method (int, int) 530
Task.AssignTaskToUsergroup Method (int, int, int) 531
DeleteTask Method 531
Task.DeleteTask Method (int) 531
Task.DeleteTask Method (int, bool) 532
DeleteTasks Method 532
Task.DeleteTasks Method (int[]) 532
Task.DeleteTasks Method (int[], bool) 533
Task.GetTask Method 533
Task.GetTasks Method 534
GetTasksAssignedByUser Method 534
Task.GetTasksAssignedByUser Method (int) 534
Task.GetTasksAssignedByUser Method (int, Ektron.Cms.Common.EkEnumeration.TaskState) 535
GetTasksAssignedToUser Method 535
Task.GetTasksAssignedToUser Method (int) 535
Task.GetTasksAssignedToUser Method (int, Ektron.Cms.Common.EkEnumeration.TaskState) 536

GetTasksAssignedToUsergroup Method 536
Task.GetTasksAssignedToUsergroup Method (int) 536
Task.GetTasksAssignedToUsergroup Method (int, Ektron.Cms.Common.EkEnumeration.TaskState) 537
Task.GetTaskState Method 537
Task.SetStateForTask Method 538
Task.UpdateTask Method 538
TaskCategory Class 538
TaskCategory Methods 539
TaskCategory.AddTaskCategory Method 540
TaskCategory.DeleteTaskCategory Method 540
TaskCategory.GetAllCategories Method 540
TaskCategory.GetTaskCategory Method 541
TaskCategory.UpdateTaskCategory Method 541
TaskCategoryType Class 542
TaskCategoryType Methods 543
TaskCategoryType.AddTaskType Method 543
TaskCategoryType.DeleteTaskType Method 543
TaskCategoryType.GetTaskType Method 544
TaskCategoryType.UpdateTaskType Method 544
Classes 545
ActiveDirectoryUser Class 545
ActiveDirectoryUser Methods 547
ActiveDirectoryUser.AddADGroupToCMS Method 548
ActiveDirectoryUser.DeSynchUserGroups Method 548
ActiveDirectoryUser.DeSynchUsers Method 548
ActiveDirectoryUser.GetADMapping Method 549
ActiveDirectoryUser.GetADStatus Method 549
ActiveDirectoryUser.GetAvailableADGroups Method 550
ActiveDirectoryUser.GetAvailableADUsers Method 550
ActiveDirectoryUser.GetCMSGroupsToSync Method 551
ActiveDirectoryUser.GetCMSRelationshipsToSync Method 551
ActiveDirectoryUser.GetCMSUsersToSync Method 551
ActiveDirectoryUser.GetDomains Method 552
ActiveDirectoryUser.MapCMSUserGroupToAD Method 552
ActiveDirectoryUser.MapCMSUserToAD Method 553
ActiveDirectoryUser.UpdateADMapping Method 553
ActiveDirectoryUser.UpdateADUsersGroups Method 553
ActiveDirectoryUser.UpdateUsersInfoFromAD Method 554

Asset Class 554
Asset Methods 556
Asset.GetAssetMgtConfigInfo Method 556
Asset.GetAssetMgtInfo Method 556
Asset.GetAssetSupertypes Method 557
Asset.GetCreateDragDropSnippet Method 557
Asset.GetCreateEktronExplorerSnippet Method 557
Asset.GetCreateSnippet Method 558
Asset.GetDragDropWOSnippet Method 558
Asset.GetEditSnippet Method 558
Asset.GetMyAssetMgtID Method 559
Asset.MakeCopyDMSAsset Method 559
Asset.SetAssetMgtConfigInfo Method 560
Blog Class 560
Blog Methods 562
Blog.AddBlog Method 562
Blog.AddCommentForPost Method 563
Blog.AddPost Method 563
Blog.DeleteBlog Method 564
Blog.DeletePost Method 564
Blog.GetBlankBlogPostData Method 564
GetBlog Method 565
Blog.GetBlog Method (int, string, int, int) 565
Blog.GetBlog Method (string, string, int, int) 566
GetBlogData Method 566
Blog.GetBlogData Method (FolderData) 567
Blog.GetBlogData Method (int) 568
Blog.GetBlogData Method (int, DateTime) 568
Blog.GetBlogData Method (int, DateTime, int, int) 569
Blog.GetBlogDataForPost Method 570
Blog.GetBlogPostData Method 570
GetBlogRoll Method 570
Blog.GetBlogRoll Method (int) 571
Blog.GetBlogRoll Method (string) 571
Blog.GetPost Method 572
BusinessRules Class 572
BusinessRules Methods 573
GetBusinessRules Method 573
BusinessRules.GetBusinessRules Method (string, string, string, string, string, string, string, int) 573

BusinessRules.GetBusinessRules Method (string, string, string, string, string, string, string, string) 574
Calendar Class 575
CalendarEvent Class 575
CalendarEventType Class 575
Content Class 576
Content Methods 578
AddContent Method 579
Content.AddContent Method (string, string, string, string, string, string, int, object, object, string) 579
Content.AddContent Method (string, string, string, string, string, string, int, object, object, string, int) 580
Content.AddContent Method (string, string, string, string, string, string, int, object, object, string, int, int) 580
Content.CheckOutContent Method 581
Content.CopyContentToFolder Method 581
Content.DeleteContentItem Method 582
Content.GetAddViewLanguage Method 582
Content.GetAllComments Method 583
Content.GetAllTemplates Method 583
Content.GetAllUnassignedItemApprovals Method 583
Content.GetChildContent Method 584
Content.GetContent Method 584
Content.GetContentByHistory Method 584
Content.GetContentForEditing Method 585
Content.GetContentState Method 585
Content.GetContentStatus Method 585
Content.GetDomain Method 586
Content.MoveContentToFolder Method 586
Content.PublishContent Method 586
Content.RenameContent Method 587
Content.SaveContent Method 587
Content.SubmitForDelete Method 588
Content.UndoCheckout Method 588
Content.UpdateContentMetaData Method 588
ContentRating Class 589
ContentRating Methods 590
ContentRating.AddContentRating Method 590
ContentRating.GetContentRatingResults Method 590
ContentRating.PurgeContentRatings Method 591
CustomFields Class 591

CustomFields Methods 593
CustomFields.AddNewEditableCustomFieldAssignments Method 594
CustomFields.BoolToYesNo Method 595
CustomFields.GetEditableCustomFieldAssignments Method 595
CustomFields.GetEnhancedMetadataArea Method 596
CustomFields.GetEnhancedMetadataScript Method 596
CustomFields.GetFieldsByFolder Method 597
CustomFields.GetSearchProperties Method 597
CustomFields.ProcessCustomFields Method 597
CustomFields.SearchAllAssets Method 598
CustomFields.WriteFilteredMetadataForEdit Method 598
CustomFields.WriteFilteredMetadataForView Method 599
CustomFields.WriteLibrarySearchExtended Method 599
CustomFields.WriteMetadataDefaultForEdit Method 599
CustomFields.WriteMetadataTypeForView Method 600
CustomFields.WritePublicSearch Method 600
Folder Class 601
Folder Methods 602
Folder.AddFolder Method 602
Folder.CreateFolder Method 602
Folder.DeleteFolder Method 603
Folder.GetChildFolders Method 603
Folder.GetFolder Method 603
GetFolderID Method 604
Folder.GetFolderID Method (int) 604
Folder.GetFolderID Method (string, int) 604
Folder.GetPath Method 605
Folder.RenameFolder Method 605
Folder.UpdateFolder Method 605
Font Class 606
Font Methods 606
Font.AddFont Method 607
Font.DeleteFont Method 607
Font.GetAllFonts Method 607
Font.GetFont Method 608
Font.GetFontConfigList Method 608
Font.UpdateFont Method 608
Form Class 609
Form Methods 610

Form.AssignFormTask Method 610
Form.EmailFormFieldData Method 611
GetAllForms Method 611
Form.GetAllForms Method () 612
Form.GetAllForms Method (int, string, int, ContentAPI.FormResultType) 612
Form.GetForm Method 612
Form.GetFormDataHistogram Method 613
GetFormFieldData Method 613
Form.GetFormFieldData Method (int, int) 613
Form.GetFormFieldData Method (int, string, string) 614
Form.GetFormFieldList Method 614
Form.GetFormFieldQuestions Method 615
Form.GetFormTitle Method 615
Form.SubmitFormFieldData Method 616
Form.UpdateFormFieldData Method 616
Library Class 616
Library Methods 618
Library.AddLoadBalanceItem Method 619
Library.DeleteLibraryItem Method 619
Library.DeleteLibraryItemById Method 619
Library.DeleteLoadBalanceItem Method 620
Library.GetAllChildLibItems Method 620
GetAllLoadBalancePaths Method 621
Library.GetAllLoadBalancePaths Method (int, string) 621
Library.GetAllLoadBalancePaths Method (string) 621
GetLibraryItem Method 622
Library.GetLibraryItem Method (int) 622
Library.GetLibraryItem Method (int, int) 622
Library.GetLibrarySettings Method 623
Library.GetLibraryTypes Method 623
Library.GetLoadBalancePath Method 624
Library.InternalLibrarySearch Method 624
Library.UpdateLibrary Method 624
Library.UpdateLibrarySettings Method 625
Library.UpdateLoadBalanceSettings Method 625
Library.UpdateQlinkTemplates Method 625
Metadata Class 626
Metadata Methods 627
Metadata.AddMetaDataType Method 627

Metadata.DeleteMetadataType Method 627
GetContentMetadataList Method 628
Metadata.GetContentMetadataList Method (int) 628
Metadata.GetContentMetadataList Method (int, bool) 628
Metadata.GetMetadataType Method 629
Metadata.GetMetaDataTypes Method 629
Metadata.UpdateMetadataType Method 630
Permissions Class 630
Permissions Methods 631
Permissions.AddItemPermission Method 632
Permissions.DeleteItemPermission Method 632
Permissions.DisableContentInheritance Method 633
Permissions.DisableFolderInheritance Method 633
Permissions.EnableContentInheritance Method 633
Permissions.EnableFolderInheritance Method 634
GetUserPermissions Method 634
Permissions.GetUserPermissions Method (int, string, int, string) 635
Permissions.GetUserPermissions Method (int, string, int, string, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) 635
Permissions.LoadPermissions Method 636
Permissions.UpdateItemPermission Method 636
SearchManager Class 636
SearchManager Methods 637
ExecuteSearch Method 637
SearchManager.ExecuteSearch Method (ContentSearchCondition) 638
SearchManager.ExecuteSearch Method (UserSearchCondition) 638
Search Method 639
SearchManager.Search Method (SearchRequestData, ref int) 639
SearchManager.Search Method (string, ref int) 640
Site Class 640
Site Methods 642
Site.BreakTaskPermissionInherit Method 642
Site.GetAllActiveLanguages Method 643
Site.GetAllLanguages Method 643
Site.GetClientDayNames Method 643
Site.GetClientMonthNames Method 644
Site.GetCookieValue Method 644
Site.GetLanguage Method 644
Site.GetLanguages Method 645

Site.GetSiteVariables Method 645
Site.GetTaskPermission Method 645
Site.IsTaskPermissionInherited Method 646
Site.SetCookieValue Method 646
Site.SetTaskPermission Method 647
Site.UpdateSiteEnabledLanguages Method 647
SiteMap Class 647
SiteMap Methods 648
SiteMap.GetFolderBreadcrumbPath Method 649
SiteMap.GetWebsiteMap Method 649
SiteMap.RenderFolderBreadcrumbHtml Method 649
SiteMap.RenderWebsiteMap Method 650
ThreadedDiscussion Class 650
ThreadedDiscussion Methods 652
ThreadedDiscussion.AddBoard Method 653
ThreadedDiscussion.AddForum Method 653
AddTopic Method 654
ThreadedDiscussion.AddTopic Method (int, string, string) 654
ThreadedDiscussion.AddTopic Method (int, string, string, int) 654
ThreadedDiscussion.DeleteBoard Method 655
ThreadedDiscussion.DeleteForum Method 655
ThreadedDiscussion.DeleteTopic Method 655
ThreadedDiscussion.GetBoardCategories Method 656
ThreadedDiscussion.GetBoardForums Method 656
GetDiscussionBoard Method 657
ThreadedDiscussion.GetDiscussionBoard Method (int) 657
ThreadedDiscussion.GetDiscussionBoard Method (string) 657
ThreadedDiscussion.GetDiscussionBoardString Method 658
ThreadedDiscussion.GetForum Method 658
ThreadedDiscussion.GetRepliesForTopics Method 658
ThreadedDiscussion.GetTopic Method 659
ThreadedDiscussion.SearchReplies Method 659
User Class 660
User Methods 663
ActivateUserAccount Method 665
User.ActivateUserAccount Method (int) 665
User.ActivateUserAccount Method (string, string) 666
User.AddMembershipUser Method 666
User.AddNewUser Method 666

User.AddUser Method 667
User.AddUserGroup Method 667
User.AddUserToGroup Method 668
User.AutoLogInUser Method 668
CreateBlankUserObject Method 668
User.CreateBlankUserObject Method () 669
User.CreateBlankUserObject Method (ref UserData) 669
User.DeleteUserByID Method 669
User.DeleteUserByIds Method 670
User.DeleteUserFromGroup Method 670
User.DeleteUserGroup Method 671
User.EditUserCustomProperties Method 671
User.GetActiveUser Method 671
User.GetADMapping Method 672
User.GetAllCmsUsers Method 672
User.GetAllCustomProperties Method 673
User.GetAllUserByCustomProperty Method 673
GetAllUserGroups Method 674
User.GetAllUserGroups Method (EkEnumeration.UserTypes, string) 674
User.GetAllUserGroups Method (string) 674
GetAllUsers Method 675
User.GetAllUsers Method (int, int) 675
User.GetAllUsers Method (ref UserRequestData) 676
User.GetGroupsUserIsIn Method 676
User.GetMembershipUser Method 676
User.GetUser Method 677
User.GetUserbyUsername Method 677
User.GetUserCustomProperties Method 678
User.GetUserEmailInfo Method 678
User.GetUserForFolderAdmin Method 679
User.GetUserGroup Method 679
User.GetUserGroupForFolderAdmin Method 679
User.GetUserPermissions Method 680
User.GetUserPreference Method 680
GetUsers Method 681
User.GetUsers Method (int, string) 681
User.GetUsers Method (string, string) 682
User.GetUsersForTask Method 682
User.GetUsersNotInGroup Method 683

IsARoleMember Method 683
User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) 684
User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, int, bool) 684
User.IsARoleMember Method (int) 685
User.IsARoleMember Method (int, int, bool) 685
User.IsARoleMemberForContent Method 686
User.IsARoleMemberForFolder Method 686
User.LogInUser Method 687
User.ResetMembershipUserPassword Method 687
User.UnsubscribeUser Method 687
User.UpdateCustomPropertiesItemOrder Method 688
UpdateUser Method 688
User.UpdateUser Method (Collection) 689
User.UpdateUser Method (UserData) 689
User.UpDateUserGroup Method 689
User.UpdateUserPreferences Method 690
UserObject Method 690
User.UserObject Method (int) 690
User.UserObject Method (string) 691

Index a

Welcome

Welcome to Ektron CMS400.NET API Online Help. This documentation helps you understand the Server Control API, Plug-in Extension API, .NET Assemblies and the Web Services API. Understanding these APIs allows you to enhance and extend your Ektron CMS400.NET site.

Copyright (c) Ektron Inc. 2006

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 Install
- http://localhost/CMS400Demo/developer/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 theperson 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 Install

http://localhost/CMS400Demo/developer/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

The Namespace containing the various Ektron classes.

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 the Visual Studio 2005 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 5)	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 5)	

Ektron.Cms.Controls Namespace

Namespaces

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

Classes

	Name	Description
	ActiveTopics (see page 9)	Summary description for Forum.
	AnalyticsTracker (see page 12)	This is class Ektron.Cms.Controls.AnalyticsTracker.
	AssetControl (see page 15)	This is class Ektron.Cms.Controls.AssetControl.
	Blog (see page 17)	Summary description for Blog.

	BlogArchive (see page 21)	The BlogCalendar (see page 24) server control.
	BlogCalendar (see page 24)	The BlogCalendar server control.
	BlogCategories (see page 27)	The BlogCategories server control.
	BlogEntries (see page 30)	Summary description for BlogEntries.
	BlogPost (see page 35)	The BlogPost server control.
	BlogRecentPosts (see page 36)	The BlogRecentPosts server control.
	BlogRoll (see page 40)	The BlogRoll server control.
	BlogRSS (see page 42)	The BlogRSS server control.
	BreadCrumb (see page 45)	The BreadCrumb server control.
	BreadCrumbItem (see page 49)	Class to contain compass item data.
	BusinessRules (see page 54)	This is class Ektron.Cms.Controls.BusinessRules.
	Calendar (see page 57)	Summary description for Calendar.
	Collection (see page 63)	Ektron Collection class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.
	ContentBlock (see page 73)	Summary description for ContentBlock.
	ContentList (see page 81)	Summary description for ContentList.
	ContentRating (see page 92)	The ContentRating server control.
	DesignTimeDiagnostic (see page 101)	This is class Ektron.Cms.Controls.DesignTimeDiagnostic.
	DhtmlMenu (see page 104)	Summary description for DhtmlMenu.
	EktronCatalogPart (see page 114)	This is class Ektron.Cms.Controls.EktronCatalogPart.
	EktronCommunityCatalogPart (see page 115)	This is class Ektron.Cms.Controls.EktronCommunityCatalogPart.
	EktronWebPartZone (see page 115)	This is class Ektron.Cms.Controls.EktronWebPartZone.
	EkWebControl (see page 117)	The base class that all Ektron CMS400 Server Controls derive from
	EkWebPart (see page 125)	This is class Ektron.Cms.Controls.EkWebPart.
	EkXsltBase (see page 125)	Summary description for EkXsltBase.
	EkXsltWebPart (see page 128)	This is class Ektron.Cms.Controls.EkXsltWebPart.
	FolderBreadcrumb (see page 128)	This control display the folder path.
	FormBlock (see page 130)	Summary description for FormBlock.
	FormFieldItem (see page 139)	Defines a field within the form.
	FormFieldItemDatalistItem (see page 142)	This is class Ektron.Cms.Controls.FormFieldItemDatalistItem.
	Forum (see page 144)	Summary description for Forum.
	HtmlEditDisplay (see page 147)	This is class Ektron.Cms.Controls.HtmlEditDisplay.
	HtmlEditor (see page 150)	This server control provides the ability to edit specific CMS content (see page 337). The eWebEditPro editor can easily be placed onto a page for content (see page 337) 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 (see page 337) and can be processed in the receiving page. [ControlName]_Action = the action to take - save, check in, undo checkout [ControlName]_ContentTitle = the content (see page 337) title [ControlName]_ContentId = the ID of the content (see page 337) [ControlName]_Language = the language of the content (see page 337) The content (see page 337) title caption... more (see page 150)
	IndexSearch (see page 167)	Summary description for IndexSearch.
	LanguageAPI (see page 187)	Control that can be used to override the default system language assignment behavior.
	LanguageSelect (see page 194)	Summary description for LanguageSelect server control.
	ListSummary (see page 196)	Summary description for ListSummary.

	Login (see page 208)	Summary description for Login.
	Membership (see page 212)	Using this control users can register to the site.
	Menu (see page 219)	Ektron Menu class.
	MetaData (see page 223)	Summary description for MetaData.
	MetaDataList (see page 225)	Ektron ContentByMeta class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.
	PersonalizationManager (see page 235)	This is class Ektron.Cms.Controls.PersonalizationManager.
	Poll (see page 242)	The Poll Server Control displays a poll or survey created from an Ektron CMS400.NET form on a Web page.
	PostHistory (see page 244)	Summary description for Forum.
	RssAggregator (see page 247)	Ektron RssAggregator class. Allows access to a RSS Feed object through IEnumerator, ICollection and IList interfaces.
	RssItemBase (see page 253)	Container for Title, Link, Description (see page 254) and DatePosted (see page 254) article information.
	Search (see page 255)	Search object returns the control and results.
	Sitemap (see page 280)	This control display the sitemap from the folder breadcrumbs.
	SmartMenu (see page 285)	Class extends the Menu-Sever-Control, adding CSS based menus.

Legend

	Class
	abstract

Ektron.Cms.Controls.Personalization Namespace

This is namespace Ektron.Cms.Controls.Personalization.

Classes

	Name	Description
	EkEditorPart (see page 7)	This is class Ektron.Cms.Controls.Personalization.EkEditorPart.

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	EkEditorPart (see page 7)	This is class Ektron.Cms.Controls.Personalization.EkEditorPart.

Legend

	Class

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.

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	ActiveTopics (see page 9)	Summary description for Forum.
	AnalyticsTracker (see page 12)	This is class Ektron.Cms.Controls.AnalyticsTracker.
	AssetControl (see page 15)	This is class Ektron.Cms.Controls.AssetControl.
	Blog (see page 17)	Summary description for Blog.
	BlogArchive (see page 21)	The BlogCalendar (see page 24) server control.
	BlogCalendar (see page 24)	The BlogCalendar server control.
	BlogCategories (see page 27)	The BlogCategories server control.
	BlogEntries (see page 30)	Summary description for BlogEntries.
	BlogPost (see page 35)	The BlogPost server control.
	BlogRecentPosts (see page 36)	The BlogRecentPosts server control.
	BlogRoll (see page 40)	The BlogRoll server control.
	BlogRSS (see page 42)	The BlogRSS server control.
	BreadCrumb (see page 45)	The BreadCrumb server control.
	BreadCrumbItem (see page 49)	Class to contain compass item data.
	BusinessRules (see page 54)	This is class Ektron.Cms.Controls.BusinessRules.
	Calendar (see page 57)	Summary description for Calendar.
	Collection (see page 63)	Ektron Collection class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.
	ContentBlock (see page 73)	Summary description for ContentBlock.
	ContentList (see page 81)	Summary description for ContentList.
	ContentRating (see page 92)	The ContentRating server control.
	DesignTimeDiagnostic (see page 101)	This is class Ektron.Cms.Controls.DesignTimeDiagnostic.
	DhtmlMenu (see page 104)	Summary description for DhtmlMenu.
	EktronCatalogPart (see page 114)	This is class Ektron.Cms.Controls.EktronCatalogPart.
	EktronCommunityCatalogPart (see page 115)	This is class Ektron.Cms.Controls.EktronCommunityCatalogPart.
	EktronWebPartZone (see page 115)	This is class Ektron.Cms.Controls.EktronWebPartZone.
	EkWebControl (see page 117)	The base class that all Ektron CMS400 Server Controls (see page 5) derive from
	EkWebPart (see page 125)	This is class Ektron.Cms.Controls.EkWebPart.
	EkXsltBase (see page 125)	Summary description for EkXsltBase.
	EkXsltWebPart (see page 128)	This is class Ektron.Cms.Controls.EkXsltWebPart.
	FolderBreadcrumb (see page 128)	This control display the folder path.
	FormBlock (see page 130)	Summary description for FormBlock.

	FormFieldItem (see page 139)	Defines a field within the form.
	FormFieldItemDatalistItem (see page 142)	This is class Ektron.Cms.Controls.FormFieldItemDatalistItem.
	Forum (see page 144)	Summary description for Forum.
	HtmlEditDisplay (see page 147)	This is class Ektron.Cms.Controls.HtmlEditDisplay.
	HtmlEditor (see page 150)	This server control provides the ability to edit specific CMS content (see page 337). The eWebEditPro editor can easily be placed onto a page for content (see page 337) 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 (see page 337) and can be processed in the receiving page. [ControlName]_Action = the action to take - save, check in, undo checkout [ControlName]_ContentTitle = the content (see page 337) title [ControlName]_ContentId = the ID of the content (see page 337) [ControlName]_Language = the language of the content (see page 337) The content (see page 337) title caption... more (see page 150)
	IndexSearch (see page 167)	Summary description for IndexSearch.
	LanguageAPI (see page 187)	Control that can be used to override the default system language assignment behavior.
	LanguageSelect (see page 194)	Summary description for LanguageSelect server control.
	ListSummary (see page 196)	Summary description for ListSummary.
	Login (see page 208)	Summary description for Login.
	Membership (see page 212)	Using this control users can register to the site.
	Menu (see page 219)	Ektron Menu class.
	MetaData (see page 223)	Summary description for MetaData.
	MetaDataList (see page 225)	Ektron ContentByMeta class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.
	PersonalizationManager (see page 235)	This is class Ektron.Cms.Controls.PersonalizationManager.
	Poll (see page 242)	The Poll Server Control displays a poll or survey created from an Ektron CMS400.NET form on a Web page.
	PostHistory (see page 244)	Summary description for Forum.
	RssAggregator (see page 247)	Ektron RssAggregator class. Allows access to a RSS Feed object through IEnumerator, ICollection and IList interfaces.
	RssItemBase (see page 253)	Container for Title, Link, Description (see page 254) and DatePosted (see page 254) article information.
	Search (see page 255)	Search object returns the control and results.
	Sitemap (see page 280)	This control display the sitemap from the folder breadcrumbs.
	SmartMenu (see page 285)	Class extends the Menu-Sever-Control, adding CSS based menus.

Legend

	Class
	abstract

ActiveTopics Class

Summary description for Forum.

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

Members

ActiveTopics Methods

	Name	Description
	Fill (see page 11)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 11)	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 11)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
ActiveTopics Methods (see page 10)	The methods of the ActiveTopics class are listed here.
ActiveTopics Properties (see page 11)	The properties of the ActiveTopics class are listed here.

ActiveTopics Properties

	Name	Description
	BoardID (see page 12)	to be changed
V	CacheInterval (see page 12)	sets the cache interval in second(s)

Legend

	Method
	Property
V	virtual

ActiveTopics Methods

	Name	Description
	Fill (see page 11)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 11)	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 11)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

ActiveTopics Properties

	Name	Description
	BoardID (see page 12)	to be changed
V	CacheInterval (see page 12)	sets the cache interval in second(s)

Legend

	Property
V	virtual

ActiveTopics Methods

The methods of the ActiveTopics class are listed here.

Public Methods

	Name	Description
	Fill (see page 11)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 11)	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 11)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
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 12)	to be changed
V	CacheInterval (see page 12)	sets the cache interval in second(s)

Legend

	Property
V	virtual

ActiveTopics.BoardID Property

C#

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

Visual Basic

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

Description

to be changed

ActiveTopics.CacheInterval Property

sets the cache interval in second(s)

C#

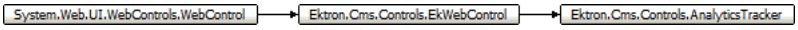
```
[Bindable(true)]
[Category("Ektron")]
[Description("interval to cache this control in second(s)")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

Visual Basic

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

AnalyticsTracker Class

Class Hierarchy



C#

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

Visual Basic

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

File

AnalyticsTracker.cs

Description

This is class Ektron.Cms.Controls.AnalyticsTracker.

Members

AnalyticsTracker Methods

	Name	Description
	Wrap (see page 14)	

Topics

Name	Description
AnalyticsTracker Methods (see page 13)	The methods of the AnalyticsTracker class are listed here.
AnalyticsTracker Properties (see page 14)	The properties of the AnalyticsTracker class are listed here.

AnalyticsTracker Properties

	Name	Description
	DefaultContentID (see page 14)	The Content (see page 337) 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 a Default Content (see page 337) ID.
	DynamicParameter (see page 14)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically - Set to "" (empty) if you wish to always display the default ContentID (static.)
	EnableAnalytics (see page 15)	Allows

Legend

	Method
	Property

AnalyticsTracker Methods

	Name	Description
	Wrap (see page 14)	

Legend

	Method

AnalyticsTracker Properties

	Name	Description
	DefaultContentID (see page 14)	The Content (see page 337) 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 a Default Content (see page 337) ID.
	DynamicParameter (see page 14)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically - Set to "" (empty) if you wish to always display the default ContentID (static.)
	EnableAnalytics (see page 15)	Allows

Legend

	Property

AnalyticsTracker Methods

The methods of the AnalyticsTracker class are listed here.

Public Methods

	Name	Description
	Wrap (see page 14)	

Legend

	Method

AnalyticsTracker.Wrap Method

C#

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

Visual Basic

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

Parameters

Parameters	Description
Html	

AnalyticsTracker Properties

The properties of the AnalyticsTracker class are listed here.

Public Properties

	Name	Description
	DefaultContentID (see page 14)	The Content (see page 337) 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 a Default Content (see page 337) ID.
	DynamicParameter (see page 14)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically - Set to "" (empty) if you wish to always display the default ContentID (static.)
	EnableAnalytics (see page 15)	Allows

Legend

	Property

AnalyticsTracker.DefaultContentID Property

The Content (see page 337) 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 a Default Content (see page 337) 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 a Default Content ID.")]
public int 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 a Default Content ID.")>
Public Property DefaultContentID() As Integer
```

AnalyticsTracker.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 337) 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("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

Allows

C#

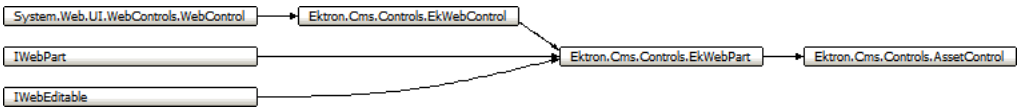
```
public ConfigOrBool EnableAnalytics;
```

Visual Basic

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

AssetControl Class

Class Hierarchy



C#

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

Visual Basic

```
Public Class AssetControl
Inherits EkWebPart
```

File

DocumentControl.cs

Description

This is class Ektron.Cms.Controls.AssetControl.

Members

AssetControl 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.

Topics

Name	Description
AssetControl Methods (see page 16)	The methods of the AssetControl class are listed here.
AssetControl Properties (see page 16)	The properties of the AssetControl class are listed here.

AssetControl Properties

	Name	Description
	DynamicParameter (see page 17)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID or Folder ID dynamically. Content (see page 337) 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.)
	UploadType (see page 17)	Selects whether the control will add a document to a folder or overwrite an existing document.

Legend

	Method
	Property

AssetControl 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.

Legend

	Method

AssetControl Properties

	Name	Description
	DynamicParameter (see page 17)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID or Folder ID dynamically. Content (see page 337) 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.)
	UploadType (see page 17)	Selects whether the control will add a document to a folder or overwrite an existing document.

Legend

	Property

AssetControl Methods

The methods of the AssetControl 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.

Legend

	Method

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 Properties

The properties of the AssetControl class are listed here.

Public Properties

	Name	Description
	DynamicParameter (see page 17)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID or Folder ID dynamically. Content (see page 337) 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.)

	UploadType (see page 17)	Selects whether the control will add a document to a folder or overwrite an existing document.

Legend

	Property

AssetControl.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 337) ID or Folder ID dynamically. Content (see page 337) 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.)

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")]
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")>
Public Property DynamicParameter() As string
```

AssetControl.UploadType Property

Selects whether the control will add a document to a folder or overwrite an existing document.

C#

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

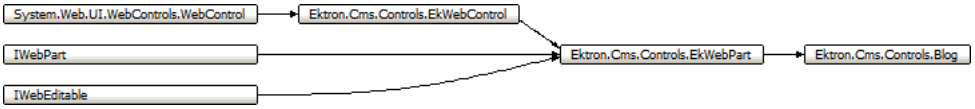
Visual Basic

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

Blog Class

Summary description for Blog.

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

Members

Blog Methods

	Name	Description
	Fill (see page 19)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 19)	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 19)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
Blog Methods (see page 19)	The methods of the Blog class are listed here.
Blog Properties (see page 20)	The properties of the Blog class are listed here.

Blog Properties

	Name	Description
	CacheInterval (see page 20)	sets the cache interval in second(s)
	PostParameter (see page 20)	Checks the QueryString for this value and shows the post when specified. Leave blank to always display the blog.
	RecentPosts (see page 20)	
	ShowHeader (see page 21)	Controls (see page 5) the display of the title and tagline.

Legend

	Method
	Property

Blog Methods

	Name	Description
	Fill (see page 19)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 19)	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 19)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

Blog Properties

	Name	Description
	CacheInterval (see page 20)	sets the cache interval in second(s)
	PostParameter (see page 20)	Checks the QueryString for this value and shows the post when specified. Leave blank to always display the blog.
	RecentPosts (see page 20)	
	ShowHeader (see page 21)	Controls (see page 5) the display of the title and tagline.

Legend

	Property

Blog Methods

The methods of the Blog class are listed here.

Public Methods

	Name	Description
	Fill (see page 19)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 19)	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 19)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
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
	CacheInterval (see page 20)	sets the cache interval in second(s)
	PostParameter (see page 20)	Checks the QueryString for this value and shows the post when specified. Leave blank to always display the blog.
	RecentPosts (see page 20)	
	ShowHeader (see page 21)	Controls (see page 5) the display of the title and tagline.

Legend

	Property

Blog.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)]
public double CacheInterval;
```

Visual Basic

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

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("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and replaces the blog with that post when
specified. Leave blank to always display the post.")]
public string PostParameter;
```

Visual Basic

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

Blog.RecentPosts Property

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Number of Recent Posts to Display")]
```

```
[DefaultValue(5)]
public int RecentPosts;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Number of Recent Posts to Display")>
<DefaultValue(5)>
Public Property RecentPosts() As Integer
```

Blog.ShowHeader Property

Controls (see page 5) the display of the title and tagline.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Controls the display of the title and tagline.")]
public bool ShowHeader;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Controls the display of the title and tagline.")>
Public Property ShowHeader() As bool
```

BlogArchive Class

The BlogCalendar (see page 24) server control.

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

Members

BlogArchive Methods

	Name	Description
	Fill (see page 22)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 23)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
BlogArchive Methods (see page 22)	The methods of the BlogArchive class are listed here.
BlogArchive Properties (see page 23)	The properties of the BlogArchive class are listed here.

BlogArchive Properties

	Name	Description
	CacheInterval (see page 23)	Sets the cache interval in second(s).
	Text (see page 23)	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 24)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Method
	Property
	virtual

BlogArchive Methods

	Name	Description
	Fill (see page 22)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 23)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogArchive Properties

	Name	Description
	CacheInterval (see page 23)	Sets the cache interval in second(s).
	Text (see page 23)	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 24)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property
	virtual

BlogArchive Methods

The methods of the BlogArchive class are listed here.

Public Methods

	Name	Description
	Fill (see page 22)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 23)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
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
	CacheInterval (see page 23)	Sets the cache interval in second(s).
	Text (see page 23)	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 24)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property
	virtual

BlogArchive.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)]
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
```

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;
```

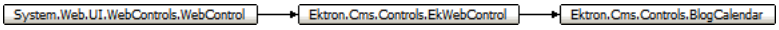
Visual Basic

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

BlogCalendar Class

The BlogCalendar server control.

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

Members

BlogCalendar Methods

	Name	Description
	Fill (see page 25)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 25)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
BlogCalendar Methods (see page 25)	The methods of the BlogCalendar class are listed here.
BlogCalendar Properties (see page 26)	The properties of the BlogCalendar class are listed here.

BlogCalendar Properties

	Name	Description
	CacheInterval (see page 26)	Sets the cache interval in second(s).
	Text (see page 26)	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 26)	Allows a developer to specify control's tag. Default is assumed " ".

Legend

	Method
	Property
	virtual

BlogCalendar Methods

	Name	Description
	Fill (see page 25)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 25)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogCalendar Properties

	Name	Description
	CacheInterval (see page 26)	Sets the cache interval in second(s).
	Text (see page 26)	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 26)	Allows a developer to specify control's tag. Default is assumed " ".

Legend

	Property
	virtual

BlogCalendar Methods

The methods of the BlogCalendar class are listed here.

Public Methods

	Name	Description
	Fill (see page 25)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 25)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
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
	CacheInterval (see page 26)	Sets the cache interval in second(s).
	Text (see page 26)	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 26)	Allows a developer to specify control's tag. Default is assumed " ".

Legend

	Property
	virtual

BlogCalendar.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)]
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.

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

Members

BlogCategories Methods

	Name	Description
	Fill (see page 28)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 28)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
BlogCategories Methods (see page 28)	The methods of the BlogCategories class are listed here.
BlogCategories Properties (see page 29)	The properties of the BlogCategories class are listed here.

BlogCategories Properties

	Name	Description
	CacheInterval (see page 29)	Sets the cache interval in second(s).
	Text (see page 29)	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 29)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Method
	Property
	virtual

BlogCategories Methods

	Name	Description
	Fill (see page 28)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 28)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogCategories Properties

	Name	Description
	CacheInterval (see page 29)	Sets the cache interval in second(s).
	Text (see page 29)	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 29)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property
	virtual

BlogCategories Methods

The methods of the BlogCategories class are listed here.

Public Methods

	Name	Description
	Fill (see page 28)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 28)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
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
	CacheInterval (see page 29)	Sets the cache interval in second(s).
	Text (see page 29)	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 29)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property
	virtual

BlogCategories.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)]
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
```

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

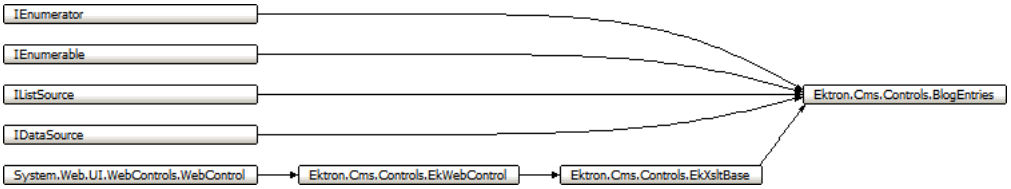
```
<Browsable(false)>
<DefaultValue("div")>
```

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

BlogEntries Class

Summary description for BlogEntries.

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

Members

BlogEntries Methods

	Name	Description
	Fill (see page 31)	Method that can be overridden and used to grab data from the back end for each server control.
	GetRssFeed (see page 32)	This method returns a blog post RSS feed.
	GetView (see page 32)	Gets the view data using dataset
	GetViewNames (see page 32)	gets the view names
	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.

Topics

Name	Description
BlogEntries Methods (see page 31)	The methods of the BlogEntries class are listed here.
BlogEntries Properties (see page 33)	The properties of the BlogEntries class are listed here.

BlogEntries Properties

	Name	Description
	BlogPostParameter (see page 33)	Checks the QueryString for this value and replaces the collection with that content (see page 337) block when specified. Leave blank to always display the collection.
	CacheInterval (see page 34)	sets the cache interval in second(s)
	DisplayXslt (see page 34)	The Xslt to use to render the Collection (see page 63) (optional)
	ShowHeader (see page 34)	Controls (see page 5) the display of the title and tagline.
	WrapTag (see page 34)	Allows a developer to specify control's tag. Default is assumed "".
	XmlDoc (see page 35)	XmlDocument used in this control.

Legend

	Method
	Property
	virtual

BlogEntries Methods

	Name	Description
	Fill (see page 31)	Method that can be overridden and used to grab data from the back end for each server control.
	GetRssFeed (see page 32)	This method returns a blog post RSS feed.
	GetView (see page 32)	Gets the view data using dataset
	GetViewNames (see page 32)	gets the view names
	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.

Legend

	Method

BlogEntries Properties

	Name	Description
	BlogPostParameter (see page 33)	Checks the QueryString for this value and replaces the collection with that content (see page 337) block when specified. Leave blank to always display the collection.
	CacheInterval (see page 34)	sets the cache interval in second(s)
	DisplayXslt (see page 34)	The Xslt to use to render the Collection (see page 63) (optional)
	ShowHeader (see page 34)	Controls (see page 5) the display of the title and tagline.
	WrapTag (see page 34)	Allows a developer to specify control's tag. Default is assumed "".
	XmlDoc (see page 35)	XmlDocument used in this control.

Legend

	Property
	virtual

BlogEntries Methods

The methods of the BlogEntries class are listed here.

Public Methods

	Name	Description
	Fill (see page 31)	Method that can be overridden and used to grab data from the back end for each server control.
	GetRssFeed (see page 32)	This method returns a blog post RSS feed.
	GetView (see page 32)	Gets the view data using dataset
	GetViewNames (see page 32)	gets the view names
	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.

Legend

	Method

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
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
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
	BlogPostParameter (see page 33)	Checks the QueryString for this value and replaces the collection with that content (see page 337) block when specified. Leave blank to always display the collection.
V	CacheInterval (see page 34)	sets the cache interval in second(s)
	DisplayXslt (see page 34)	The Xslt to use to render the Collection (see page 63) (optional)
	ShowHeader (see page 34)	Controls (see page 5) the display of the title and tagline.
	WrapTag (see page 34)	Allows a developer to specify control's tag. Default is assumed "".
	XmlDoc (see page 35)	XmlDocument used in this control.

Legend

	Property
V	virtual

BlogEntries.BlogPostParameter Property

Checks the QueryString for this value and replaces the collection with that content (see page 337) 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 blog entries with that
post when specified. Leave blank to always display the blog entries.")]
public string BlogPostParameter;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and replaces the blog entries with that
```

```
post when specified. Leave blank to always display the blog entries.")>
Public Property BlogPostParameter() As string
```

BlogEntries.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)]
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
```

BlogEntries.DisplayXslt Property

The Xslt to use to render the Collection (see page 63) (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.ShowHeader Property

Controls (see page 5) the display of the title and tagline.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Controls the display of the title and tagline.")]
public bool ShowHeader;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Controls the display of the title and tagline.")>
Public Property ShowHeader() As bool
```

BlogEntries.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.XmlDoc Property

XmlDocument used in this control.

C#

```
[Browsable(false)]
public override XmlDocument XmlDoc;
```

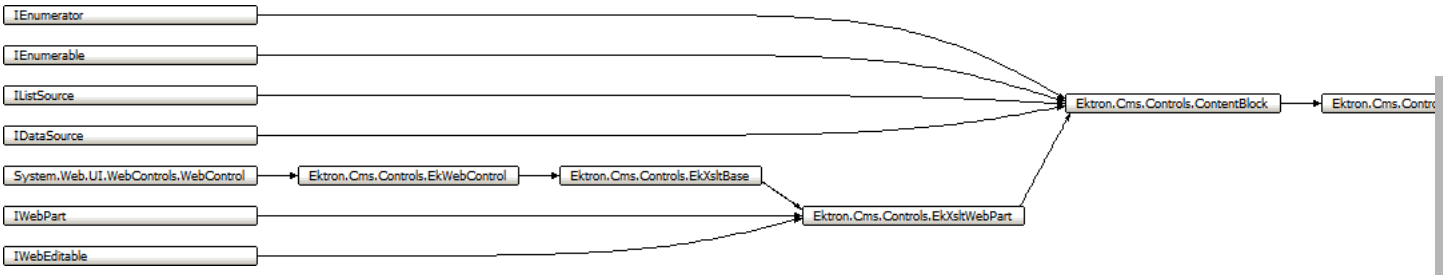
Visual Basic

```
<Browsable(false)>
Public override Property XmlDoc() As XmlDocument
```

BlogPost Class

The BlogPost server control.

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

Members

BlogPost Methods

	Name	Description
	Wrap (see page 36)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
BlogPost Methods (see page 36)	The methods of the BlogPost class are listed here.

Legend

	Method

BlogPost Methods

	Name	Description
	Wrap (see page 36)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogPost Methods

The methods of the BlogPost class are listed here.

Public Methods

	Name	Description
	Wrap (see page 36)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

BlogRecentPosts Class

The BlogRecentPosts server control.

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

Members

BlogRecentPosts Methods

	Name	Description
	Fill (see page 38)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 38)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
BlogRecentPosts Methods (see page 38)	The methods of the BlogRecentPosts class are listed here.
BlogRecentPosts Properties (see page 38)	The properties of the BlogRecentPosts class are listed here.

BlogRecentPosts Properties

	Name	Description
V	CacheInterval (see page 39)	Sets the cache interval in second(s).
	NumberofPosts (see page 39)	
	Text (see page 39)	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 39)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Method
	Property
V	virtual

BlogRecentPosts Methods

	Name	Description
	Fill (see page 38)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 38)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogRecentPosts Properties

	Name	Description
V	CacheInterval (see page 39)	Sets the cache interval in second(s).
	NumberofPosts (see page 39)	
	Text (see page 39)	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 39)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property
V	virtual

BlogRecentPosts Methods

The methods of the BlogRecentPosts class are listed here.

Public Methods

	Name	Description
	Fill (see page 38)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 38)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
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
	CacheInterval (see page 39)	Sets the cache interval in second(s).
	NumberofPosts (see page 39)	
	Text (see page 39)	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 39)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property
	virtual

BlogRecentPosts.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)]
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
```

BlogRecentPosts.NumberofPosts Property

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Number of Recent Posts to Display")]
[DefaultValue(5)]
public int NumberofPosts;
```

Visual Basic

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

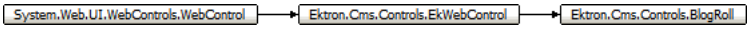
Visual Basic

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

BlogRoll Class

The BlogRoll server control.

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

Members

BlogRoll Methods

	Name	Description
	Fill (see page 41)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 41)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
BlogRoll Methods (see page 41)	The methods of the BlogRoll class are listed here.
BlogRoll Properties (see page 41)	The properties of the BlogRoll class are listed here.

BlogRoll Properties

	Name	Description
	CacheInterval (see page 42)	Sets the cache interval in second(s).
	Text (see page 42)	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 "".

Legend

	Method
	Property
	virtual

BlogRoll Methods

	Name	Description
	Fill (see page 41)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 41)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogRoll Properties

	Name	Description
	CacheInterval (see page 42)	Sets the cache interval in second(s).
	Text (see page 42)	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 "".

Legend

	Property
	virtual

BlogRoll Methods

The methods of the BlogRoll class are listed here.

Public Methods

	Name	Description
	Fill (see page 41)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 41)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
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
	CacheInterval (see page 42)	Sets the cache interval in second(s).
	Text (see page 42)	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 "".

Legend

	Property
	virtual

BlogRoll.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)]
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
```

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;
```

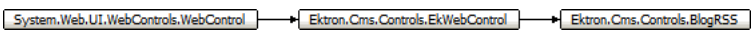
Visual Basic

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

BlogRSS Class

The BlogRSS server control.

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

Members

BlogRSS Methods

	Name	Description
	Fill (see page 44)	Overrideable method that can be used to grab data from the back end for each server control.

Topics

Name	Description
BlogRSS Methods (see page 44)	The methods of the BlogRSS class are listed here.
BlogRSS Properties (see page 44)	The properties of the BlogRSS class are listed here.

BlogRSS Properties

	Name	Description
	CacheInterval (see page 44)	Sets the cache interval in second(s).
	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 45)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Method
	Property
	virtual

BlogRSS Methods

	Name	Description
	Fill (see page 44)	Overrideable method that can be used to grab data from the back end for each server control.

Legend

	Method

BlogRSS Properties

	Name	Description
	CacheInterval (see page 44)	Sets the cache interval in second(s).
	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 45)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property
	virtual

BlogRSS Methods

The methods of the BlogRSS class are listed here.

Public Methods

	Name	Description
	Fill (see page 44)	Overrideable method that can be used to grab data from the back end for each server control.

Legend

	Method

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
	CacheInterval (see page 44)	Sets the cache interval in second(s).
	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 45)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property
	virtual

BlogRSS.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)]
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
```

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

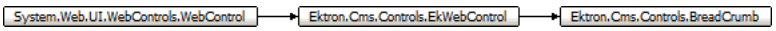
Visual Basic

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

BreadCrumb Class

The BreadCrumb server control.

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

Members

BreadCrumb 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.
	GenerateTrail_HTML (see page 47)	Generates Trail HTML for both vertical and horizontal styles This demonstrates usage of HtmlTextWriter
	ResetTrail (see page 47)	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

Topics

Name	Description
BreadCrumb Methods (see page 46)	The methods of the BreadCrumb class are listed here.
BreadCrumb Properties (see page 48)	The properties of the BreadCrumb class are listed here.

BreadCrumb Properties

	Name	Description
	DynamicParameter (see page 48)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically - Set to String.Empty if you wish to always display the default ContentID (static.)
	MetadataName (see page 49)	Gets or Sets the Default ContentID for this Control.

Legend

	Method
	Property

BreadCrumb 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.
	GenerateTrail_HTML (see page 47)	Generates Trail HTML for both vertical and horizontal styles This demonstrates usage of HtmlTextWriter
	ResetTrail (see page 47)	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

Legend

	Method

BreadCrumb Properties

	Name	Description
	DynamicParameter (see page 48)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically - Set to String.Empty if you wish to always display the default ContentID (static.)
	MetadataName (see page 49)	Gets or Sets the Default ContentID for this Control.

Legend

	Property

BreadCrumb Methods

The methods of the BreadCrumb 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.
	GenerateTrail_HTML (see page 47)	Generates Trail HTML for both vertical and horizontal styles This demonstrates usage of HtmlTextWriter
	ResetTrail (see page 47)	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

Legend

	Method

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 HtmlTextWriter

C#

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

Visual Basic

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

Parameters

Parameters	Description
output	HtmlTextWriter to write into

ResetTrail Method

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 47)	Resets and removes all current Trail items.
	BreadCrumb.ResetTrail (ArrayList) (see page 48)	Resets the Trail with an array list of new Trail items.
	BreadCrumb.ResetTrail (BreadCrumbItem) (see page 48)	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
TrailData	

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
item	

BreadCrumb Properties

The properties of the BreadCrumb class are listed here.

Public Properties

	Name	Description
	DynamicParameter (see page 48)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically - Set to String.Empty if you wish to always display the default ContentID (static.)
	MetadataName (see page 49)	Gets or Sets the Default ContentID for this Control.

Legend

	Property

BreadCrumb.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 337) 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.MetadataName Property

Gets or Sets the Default ContentID for this Control.

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
```

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

Members

Public Methods

	Name	Description
	BreadCrumbItem (see page 51)	

BreadCrumbItem Methods

	Name	Description
	Equals (see page 52)	Overridden equals function returns false on null object comparison. And, when the given object and current object .ToString (see page 52)() do not match, The ToString (see page 52)() function is also overridden to allow this to work.
	GetHashCode (see page 52)	Overridden otherwise VS.NET gives a warning.
	ToString (see page 52)	Returns a concatenated string of all class variables.ToString().

Topics

Name	Description
BreadCrumbItem Methods (see page 51)	The methods of the BreadCrumbItem class are listed here.
BreadCrumbItem Properties (see page 52)	The properties of the BreadCrumbItem class are listed here.

BreadCrumbItem Properties

	Name	Description
S	CaseSensitiveUrlComparison (see page 53)	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 53)	Gets / Sets the title for use with the display in the compass.
	IconAlt (see page 53)	Gets / Sets the text to use as the item alt text.
	IconPath (see page 53)	Gets / Sets the root relative path to the item icon.
	RawUrl (see page 53)	Gets / Sets the URL to use for the item link.
	Target (see page 54)	Gets / Sets the target of the item.

Legend

	Constructor
	Property
S	static

BreadCrumbItem Methods

	Name	Description
	Equals (see page 52)	Overridden equals function returns false on null object comparison. And, when the given object and current object .ToString (see page 52)() do not match, The ToString (see page 52)() function is also overridden to allow this to work.
	GetHashCode (see page 52)	Overridden otherwise VS.NET gives a warning.
	ToString (see page 52)	Returns a concatenated string of all class variables.ToString().

Legend

	Method

BreadCrumbItem Properties

	Name	Description
S	CaseSensitiveUrlComparison (see page 53)	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 53)	Gets / Sets the title for use with the display in the compass.
	IconAlt (see page 53)	Gets / Sets the text to use as the item alt text.
	IconPath (see page 53)	Gets / Sets the root relative path to the item icon.
	RawUrl (see page 53)	Gets / Sets the URL to use for the item link.
	Target (see page 54)	Gets / Sets the target of the item.

Legend

	Property
S	static

Public Methods

	Name	Description
	BreadCrumbItem (see page 51)	

Legend

	Constructor

BreadCrumbItem Constructor

Overload List

	Name	Description
	BreadCrumbItem.BreadCrumbItem () (see page 51)	Default no parameter constructor. This is overloaded. See Also: BreadCrumbItem.BreadCrumbItem Constructor (string, string, string, string, string) (see page 51).
	BreadCrumbItem.BreadCrumbItem (string, string, string, string, string) (see page 51)	All parameter constructor. This is overloaded. See Also: BreadCrumbItem.BreadCrumbItem Constructor () (see page 51)

BreadCrumbItem.BreadCrumbItem Constructor ()

Default no parameter constructor.

This is overloaded. See Also: BreadCrumbItem.BreadCrumbItem Constructor (string, string, string, string, string) (see page 51).

C#

```
public BreadCrumbItem();
```

Visual Basic

```
Public Sub BreadCrumbItem()
```

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

All parameter constructor.

This is overloaded. See Also: BreadCrumbItem.BreadCrumbItem Constructor () (see page 51)

C#

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

Visual Basic

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

Parameters

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

BreadCrumbItem Methods

The methods of the BreadCrumbItem class are listed here.

Public Methods

	Name	Description
	Equals (see page 52)	Overridden equals function returns false on null object comparison. And, when the given object and current object .ToString (see page 52)() do not match, The ToString (see page 52)() function is also overridden to allow this to work.
	GetHashCode (see page 52)	Overridden otherwise VS.NET gives a warning.

	ToString (see page 52)	Returns a concatenated string of all class variables.ToString().

Legend

	Method

BreadCrumbItem.Equals Method

Overridden equals function returns false on null object comparison. And, when the given object and current object .ToString (see page 52)() do not match, The ToString (see page 52)() 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
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 53)	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 53)	Gets / Sets the title for use with the display in the compass.
	IconAlt (see page 53)	Gets / Sets the text to use as the item alt text.
	IconPath (see page 53)	Gets / Sets the root relative path to the item icon.

	RawUrl (see page 53)	Gets / Sets the URL to use for the item link.
	Target (see page 54)	Gets / Sets the target of the item.

Legend

	Property
S	static

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

This is class Ektron.Cms.Controls.BusinessRules.

Members

BusinessRules Methods

	Name	Description
	Fill (see page 55)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 55)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
BusinessRules Methods (see page 55)	The methods of the BusinessRules class are listed here.
BusinessRules Properties (see page 56)	The properties of the BusinessRules class are listed here.

BusinessRules Properties

	Name	Description
	Parameters (see page 56)	User Defined parameters that can be evaluated in rules.
	Result (see page 56)	An array of boolean values representing the results of the evaluations.
	Text (see page 56)	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.

Legend

	Method
	Property

BusinessRules Methods

	Name	Description
	Fill (see page 55)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 55)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BusinessRules Properties

	Name	Description
	Parameters (see page 56)	User Defined parameters that can be evaluated in rules.
	Result (see page 56)	An array of boolean values representing the results of the evaluations.
	Text (see page 56)	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.

Legend

	Property

BusinessRules Methods

The methods of the BusinessRules class are listed here.

Public Methods

	Name	Description
	Fill (see page 55)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 55)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
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 56)	User Defined parameters that can be evaluated in rules.
	Result (see page 56)	An array of boolean values representing the results of the evaluations.
	Text (see page 56)	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.

Legend

	Property

BusinessRules.Parameters Property

User 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.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;
```

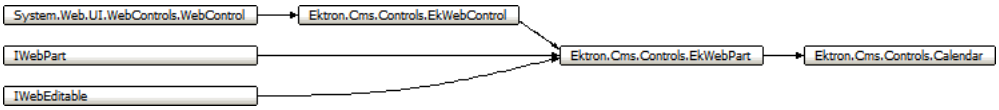
Visual Basic

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

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

Members

Calendar Methods

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

Topics

Name	Description
Calendar Methods (see page 58)	The methods of the Calendar class are listed here.
Calendar Properties (see page 59)	The properties of the Calendar class are listed here.

Calendar Properties

	Name	Description
	DefaultCalendarID (see page 60)	The default ID of the Calendar to be displayed.
	DisplayType (see page 60)	Specify the the Calendar view. Choices are: • "Month" • "Day" • "Inline" • "MonthUpWithEvent"
	DynamicParameter (see page 60)	The QueryString to check for a dynamic ID.
	EndDate (see page 61)	Specify the end date the Calendar displays.

	HighlightEventID (see page 61)	Specify the event ID which the Calendar will highlight.
	HighlightEventTypeID (see page 61)	Specify the event type and the Calendar highlights that type of event(s).
	ShowXML (see page 62)	Text property will hold raw calendar (see page 336) XML. The Calendar is not visible at runtime.
	StartDate (see page 62)	Specify the start date the calendar (see page 336) displays.
	Text (see page 62)	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.

Legend

	Method
	Property

Calendar Methods

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

Legend

	Method

Calendar Properties

	Name	Description
	DefaultCalendarID (see page 60)	The default ID of the Calendar to be displayed.
	DisplayType (see page 60)	Specify the the Calendar view. Choices are: • "Month" • "Day" • "Inline" • "MonthUpWithEvent"
	DynamicParameter (see page 60)	The QueryString to check for a dynamic ID.
	EndDate (see page 61)	Specify the end date the Calendar displays.
	HighlightEventID (see page 61)	Specify the event ID which the Calendar will highlight.
	HighlightEventTypeID (see page 61)	Specify the event type and the Calendar highlights that type of event(s).
	ShowXML (see page 62)	Text property will hold raw calendar (see page 336) XML. The Calendar is not visible at runtime.
	StartDate (see page 62)	Specify the start date the calendar (see page 336) displays.
	Text (see page 62)	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.

Legend

	Property

Calendar Methods

The methods of the Calendar class are listed here.

Public Methods

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

Legend

	Method

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
`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 60)	The default ID of the Calendar (see page 57) to be displayed.
	DisplayType (see page 60)	Specify the the Calendar (see page 57) view. Choices are: • "Month" • "Day" • "Inline" • "MonthUpWithEvent"
	DynamicParameter (see page 60)	The QueryString to check for a dynamic ID.
	EndDate (see page 61)	Specify the end date the Calendar (see page 57) displays.
	HighlightEventID (see page 61)	Specify the event ID which the Calendar (see page 57) will highlight.
	HighlightEventTypeID (see page 61)	Specify the event type and the Calendar (see page 57) highlights that type of event(s).
	ShowXML (see page 62)	Text property will hold raw calendar (see page 336) XML. The Calendar (see page 57) is not visible at runtime.
	StartDate (see page 62)	Specify the start date the calendar (see page 336) displays.
	Text (see page 62)	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.

Legend

	Property

Calendar.DefaultCalendarID Property

The default ID of the Calendar (see page 57) 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 int 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 Integer
```

Calendar.DisplayType Property

Specify the the Calendar (see page 57) 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 57) 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 57) will highlight.

C#

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

Visual Basic

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

Calendar.HighlightEventTypeID Property

Specify the event type and the Calendar (see page 57) 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 int 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 Integer
```

Calendar.ShowXML Property

Text property will hold raw calendar (see page 336) XML. The Calendar (see page 57) 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 (see page 336) 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;
```

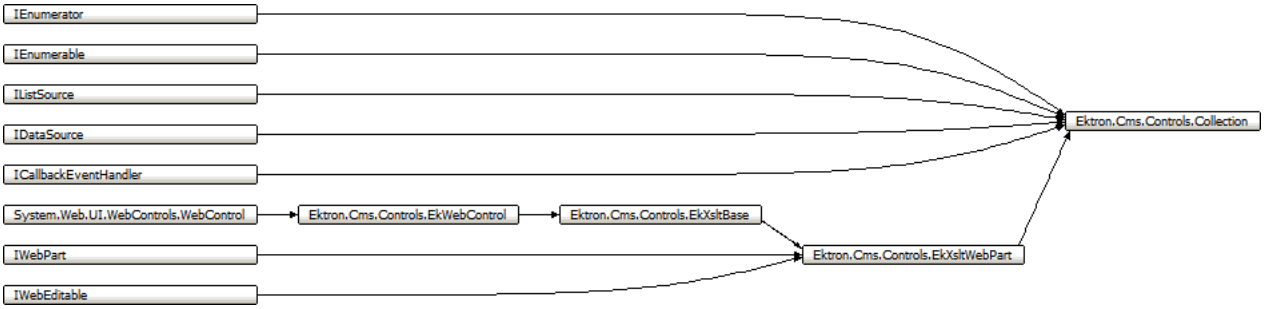
Visual Basic

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

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

Members

Collection Methods

	Name	Description
	CopyTo (see page 66)	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 66)	Calls the database and updates the contents of the collection if the data is out of sync with the specified parameters.
	GetEnumerator (see page 66)	Gets a System.Collections.IEnumerator for the collection.
	GetRssFeed (see page 66)	This method returns a Collection RSS feed.
	GetView (see page 66)	Gets the view data using a dataset.
	GetViewNames (see page 67)	Gets the view names.
	MoveNext (see page 67)	Moves up the index of the Collection. Returns false if the end of the collection has been surpassed.
	Reset (see page 67)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 67)	Index operator into a strongly typed dataset holding some common properties for a Collection item.
	ToString (see page 67)	Wraps the resulting collection with the Ektron toolbars, which will only display when a user is logged in.
	Wrap (see page 68)	Overrideable method that's used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
Collection Methods (see page 65)	The methods of the Collection class are listed here.
Collection Properties (see page 68)	The properties of the Collection class are listed here.

Collection Properties

	Name	Description
	CacheInterval (see page 68)	Sets the cache interval in second(s).
	ContentParameter (see page 69)	Checks the QueryString for this value and replaces the collection with that content (see page 337) block when specified. Leave blank to always display the collection.
	Count (see page 69)	Indicates the number of elements in your Collection.
	Current (see page 69)	Gets the current object in the collection.
	DefaultCollectionID (see page 70)	Gets or Sets the Default CollectionID for this Control.
	DynamicParameter (see page 70)	The parameter to query for a Dynamic Collection.
	EkItems (see page 70)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection. Holds more information than the exposed strongly typed this[i] index operator.
	EnablePaging (see page 71)	Enables paging if the total number of records exceeds the MaxResult value.
	GetHtml (see page 71)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	IncludeIcons (see page 71)	Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.
	IsSynchronized (see page 72)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 72)	Gets or Sets the target window where the result are displayed.
	MaxResults (see page 72)	Maximum number of results to be returned.
	Random (see page 72)	If set to true, grabs a single content (see page 337) block randomly from the collection. Set to false for all data in the collection.
	SyncRoot (see page 73)	Gets an object that's used to synchronize the collection.
	XmlDoc (see page 73)	Returns the Xslt object for the collection.

Legend

	Method
	Property
	virtual

Collection Methods

	Name	Description
	CopyTo (see page 66)	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 66)	Calls the database and updates the contents of the collection if the data is out of sync with the specified parameters.
	GetEnumerator (see page 66)	Gets a System.Collections.IEnumerator for the collection.
	GetRssFeed (see page 66)	This method returns a Collection RSS feed.
	GetView (see page 66)	Gets the view data using a dataset.
	GetViewNames (see page 67)	Gets the view names.
	MoveNext (see page 67)	Moves up the index of the Collection. Returns false if the end of the collection has been surpassed.
	Reset (see page 67)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 67)	Index operator into a strongly typed dataset holding some common properties for a Collection item.
	ToString (see page 67)	Wraps the resulting collection with the Ektron toolbars, which will only display when a user is logged in.
	Wrap (see page 68)	Overrideable method that's used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

Collection Properties

	Name	Description
	CacheInterval (see page 68)	Sets the cache interval in second(s).
	ContentParameter (see page 69)	Checks the QueryString for this value and replaces the collection with that content (see page 337) block when specified. Leave blank to always display the collection.
	Count (see page 69)	Indicates the number of elements in your Collection.
	Current (see page 69)	Gets the current object in the collection.
	DefaultCollectionID (see page 70)	Gets or Sets the Default CollectionID for this Control.
	DynamicParameter (see page 70)	The parameter to query for a Dynamic Collection.
	EkItems (see page 70)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection. Holds more information than the exposed strongly typed this[i] index operator.
	EnablePaging (see page 71)	Enables paging if the total number of records exceeds the MaxResult value.
	GetHtml (see page 71)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	IncludeIcons (see page 71)	Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.
	IsSynchronized (see page 72)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 72)	Gets or Sets the target window where the result are displayed.
	MaxResults (see page 72)	Maximum number of results to be returned.
	Random (see page 72)	If set to true, grabs a single content (see page 337) block randomly from the collection. Set to false for all data in the collection.
	SyncRoot (see page 73)	Gets an object that's used to synchronize the collection.
	XmlDoc (see page 73)	Returns the Xslt object for the collection.

Legend

	Property
	virtual

Collection Methods

The methods of the Collection class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 66)	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 66)	Calls the database and updates the contents of the collection if the data is out of sync with the specified parameters.
	GetEnumerator (see page 66)	Gets a System.Collections.IEnumerator for the collection.
	GetRssFeed (see page 66)	This method returns a Collection (see page 63) RSS feed.
	GetView (see page 66)	Gets the view data using a dataset.
	GetViewNames (see page 67)	Gets the view names.
	MoveNext (see page 67)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 67)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 67)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	ToString (see page 67)	Wraps the resulting collection with the Ektron toolbars, which will only display when a user is logged in.
	Wrap (see page 68)	Overrideable method that's used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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 63) 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
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 63). 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 63) 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
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
	CacheInterval (see page 68)	Sets the cache interval in second(s).
	ContentParameter (see page 69)	Checks the QueryString for this value and replaces the collection with that content (see page 337) block when specified. Leave blank to always display the collection.
	Count (see page 69)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 69)	Gets the current object in the collection.
	DefaultCollectionID (see page 70)	Gets or Sets the Default CollectionID for this Control.
	DynamicParameter (see page 70)	The parameter to query for a Dynamic Collection (see page 63).
	EkItems (see page 70)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	EnablePaging (see page 71)	Enables paging if the total number of records exceeds the MaxResult value.
	GetHtml (see page 71)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	IncludeIcons (see page 71)	Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.
	IsSynchronized (see page 72)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 72)	Gets or Sets the target window where the result are displayed.
	MaxResults (see page 72)	Maximum number of results to be returned.
	Random (see page 72)	If set to true, grabs a single content (see page 337) block randomly from the collection. Set to false for all data in the collection.
	SyncRoot (see page 73)	Gets an object that's used to synchronize the collection.
	XmlDoc (see page 73)	Returns the Xslt object for the collection.

Legend

	Property
	virtual

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()]
public virtual double CacheInterval;
```

Visual Basic

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

Collection.ContentParameter Property

Checks the QueryString for this value and replaces the collection with that content (see page 337) 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()]
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()>
Public Property ContentParameter() As string
```

Collection.Count Property

Indicates the number of elements in your Collection (see page 63).

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()]
public int 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()>
Public Property DefaultCollectionID() As Integer
```

Collection.DynamicParameter Property

The parameter to query for a Dynamic Collection (see page 63).

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[Description("The parameter to query for a Dynamic Collection.")]
[TypeConverter(typeof(DynamicCollectionTypeConverter))]
[Personalizable()]
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()>
Public Property DynamicParameter() As string
```

Collection.EkItems Property

Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). 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()]
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()>
Public Property EnablePaging() As bool
```

Collection.GetHtml Property

If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) 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()]
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()>
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()]
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()>
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

Gets or Sets the target window where the result are 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
```

Collection.MaxResults Property

Maximum number of results to be returned.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[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(0)>
<Description("The maximum number of results to fetch. If set to zero, there is no limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

Collection.Random Property

If set to true, grabs a single content (see page 337) 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()]
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()>
Public Property Random() 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;
```

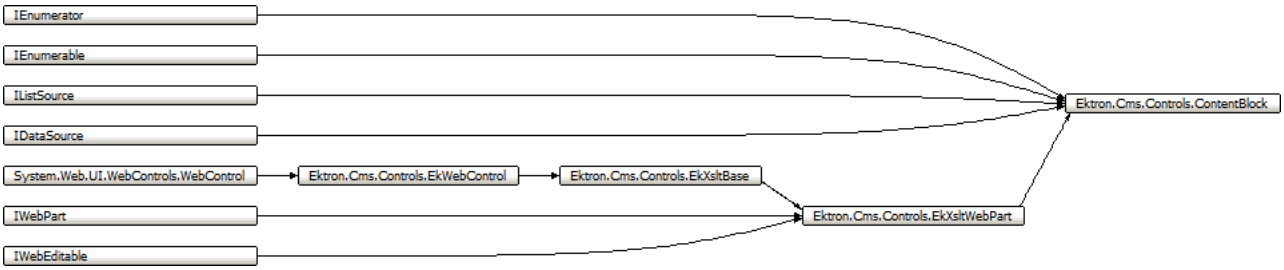
Visual Basic

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

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;
```

Visual Basic

```
<DefaultProperty("DefaultContentID"), ToolboxData("<{0}:ContentBlock
runat=server></{0}:ContentBlock>")>
Public Class ContentBlock
Inherits EkXsltWebPart
Inherits IEnumerator
Inherits IEnumerable
```

```
Inherits IListSource
Inherits IDataSource
```

File

ContentBlock.cs

Members

ContentBlock Methods

	Name	Description
	CopyTo (see page 76)	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 76)	Method that can be overridden and used to grab data from the back end for each server control.
	GetEnumerator (see page 76)	Gets a System.Collections.IEnumerator for the collection.
	MoveNext (see page 76)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 76)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 77)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	ToString (see page 77)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	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.

Topics

Name	Description
ContentBlock Methods (see page 75)	The methods of the ContentBlock class are listed here.
ContentBlock Properties (see page 77)	The properties of the ContentBlock class are listed here.

ContentBlock Properties

	Name	Description
	AllowMembershipEditing (see page 78)	Set to true to allow membership editors, with sufficient editing permissions, to edit content (see page 337). Defaults to false.
V	CacheInterval (see page 78)	Sets the cache interval in second(s).
	Count (see page 78)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 79)	Gets the current object in the collection.
	DisplayXslt (see page 79)	Gets or Sets the path to an Xslt that can be applied to this content (see page 337) block. Set to String.Empty if you wish to always display the default Content (see page 337) view.
	DynamicParameter (see page 79)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)
	EkItem (see page 80)	Gets the Ektron.Cms.Common.ContentBase for the current Content (see page 337) Block.
	IsSynchronized (see page 80)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	OverrideXslt (see page 80)	Ektron CMS400.NET allows an Administrator to specify up to three custom Xslt's which can be applied to a content (see page 337) block from the System. This allows the user to override the default ContentBlock 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 (see page 337) block (see Xslt property.)
	StartingFolderID (see page 80)	For use with Web Parts only. Allows a developer to set the starting folder for the personalization/collaboration selection dialog.
	SyncRoot (see page 81)	Gets an object that can be used to synchronize the collection.

Legend

	Method
	Property
V	virtual

ContentBlock Methods

	Name	Description
	CopyTo (see page 76)	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 76)	Method that can be overridden and used to grab data from the back end for each server control.
	GetEnumerator (see page 76)	Gets a System.Collections.IEnumerator for the collection.
	MoveNext (see page 76)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 76)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 77)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	ToString (see page 77)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	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.

Legend

	Method

ContentBlock Properties

	Name	Description
	AllowMembershipEditing (see page 78)	Set to true to allow membership editors, with sufficient editing permissions, to edit content (see page 337). Defaults to false.
V	CacheInterval (see page 78)	Sets the cache interval in second(s).
	Count (see page 78)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 79)	Gets the current object in the collection.
	DisplayXslt (see page 79)	Gets or Sets the path to an Xslt that can be applied to this content (see page 337) block. Set to String.Empty if you wish to always display the default Content (see page 337) view.
	DynamicParameter (see page 79)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)
	EkItem (see page 80)	Gets the Ektron.Cms.Common.ContentBase for the current Content (see page 337) Block.
	IsSynchronized (see page 80)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	OverrideXslt (see page 80)	Ektron CMS400.NET allows an Administrator to specify up to three custom Xslt's which can be applied to a content (see page 337) block from the System. This allows the user to override the default ContentBlock 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 (see page 337) block (see Xslt property.)
	StartingFolderID (see page 80)	For use with Web Parts only. Allows a developer to set the starting folder for the personalization/collaboration selection dialog.
	SyncRoot (see page 81)	Gets an object that can be used to synchronize the collection.

Legend

	Property
V	virtual

ContentBlock Methods

The methods of the ContentBlock class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 76)	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 76)	Method that can be overridden and used to grab data from the back end for each server control.
	GetEnumerator (see page 76)	Gets a System.Collections.IEnumerator for the collection.
	MoveNext (see page 76)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 76)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 77)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	ToString (see page 77)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	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.

Legend

	Method

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 63). 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 63) 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
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
	AllowMembershipEditing (see page 78)	Set to true to allow membership editors, with sufficient editing permissions, to edit content (see page 337). Defaults to false.
	CacheInterval (see page 78)	Sets the cache interval in second(s).
	Count (see page 78)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 79)	Gets the current object in the collection.
	DisplayXslt (see page 79)	Gets or Sets the path to an Xslt that can be applied to this content (see page 337) block. Set to String.Empty if you wish to always display the default Content (see page 337) view.
	DynamicParameter (see page 79)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)
	EkItem (see page 80)	Gets the Ektron.Cms.Common.ContentBase for the current Content (see page 337) Block.
	IsSynchronized (see page 80)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	OverrideXslt (see page 80)	Ektron CMS400.NET allows an Administrator to specify up to three custom Xslt's which can be applied to a content (see page 337) block from the System. This allows the user to override the default ContentBlock (see page 73) 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 (see page 337) block (see Xslt property.)

	StartingFolderID (see page 80)	For use with Web Parts only. Allows a developer to set the starting folder for the personalization/collaboration selection dialog.
	SyncRoot (see page 81)	Gets an object that can be used to synchronize the collection.

Legend

	Property
	virtual

ContentBlock.AllowMembershipEditing Property

Set to true to allow membership editors, with sufficient editing permissions, to edit content (see page 337). Defaults to false.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to true to allow membership editors, with sufficient editing permissions, to
edit content. Defaults to false.")]
[Personalizable()]
public bool AllowMembershipEditing;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to allow membership editors, with sufficient editing permissions, to
edit content. Defaults to false.")>
<Personalizable()>
Public Property AllowMembershipEditing() As bool
```

ContentBlock.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()]
public virtual double CacheInterval;
```

Visual Basic

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

ContentBlock.Count Property

Indicates the number of elements in your Collection (see page 63).

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.DisplayXslt Property

Gets or Sets the path to an Xslt that can be applied to this content (see page 337) block. Set to String.Empty if you wish to always display the default Content (see page 337) view.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("id")]
[TypeConverter("DynamicParameterTypeConverter")]
[Description("Use this to specify an XSLT that will override the default display for an XML
content block.")]
[Personalizable()]
public string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("id")>
<TypeConverter("DynamicParameterTypeConverter")>
<Description("Use this to specify an XSLT that will override the default display for an XML
content block.")>
<Personalizable()>
Public Property DisplayXslt() As string
```

ContentBlock.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 337) 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()]
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()>
Public Property DynamicParameter() As string
```

ContentBlock.EkItem Property

Gets the Ektron.Cms.Common.ContentBase for the current Content (see page 337) 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 (see page 337) block from the System. This allows the user to override the default ContentBlock (see page 73) 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 (see page 337) block (see Xslt property.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("id")]
[Description("Use this to specify an XSLT that will override the default display for an XML
content block.")]
public XsltOverride OverrideXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("id")>
<Description("Use this to specify an XSLT that will override the default display for an XML
content block.")>
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()]
public int 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 Integer
```

ContentBlock.SyncRoot Property

Gets an object that can be used to synchronize the collection.

C#

```
[Browsable(false)]
public object SyncRoot;
```

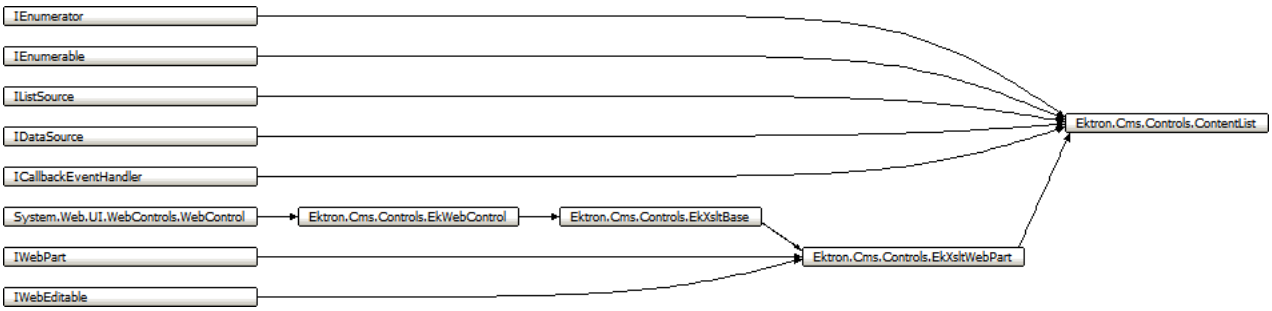
Visual Basic

```
<Browsable(false)>
Public Property SyncRoot() As object
```

ContentList Class

Summary description for ContentList.

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

Members

ContentList Methods

	Name	Description
	CopyTo (see page 84)	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 84)	Populates the data for the ContentList.
	GetEnumerator (see page 84)	Gets a System.Collections.IEnumerator for the ContentList.
	GetRssFeed (see page 84)	This method returns the ContentList RSS feed.
	MoveNext (see page 85)	Move to next index.
	Reset (see page 85)	Resets the index.
	this (see page 85)	Returns the current ContentItem.ContentRow.
	ToString (see page 85)	Overrides the system ToString function.
	Wrap (see page 86)	Wrap the Html toolbar.

Topics

Name	Description
ContentList Methods (see page 83)	The methods of the ContentList class are listed here.
ContentList Properties (see page 86)	The properties of the ContentList class are listed here.

ContentList Properties

	Name	Description
V	CacheInterval (see page 86)	Sets the cache interval in second(s).
	ContentIds (see page 87)	The comma separated list of content (see page 337) IDs you wish to display.
	Count (see page 87)	Indicates the number of elements in your ContentList.
	Current (see page 87)	Gets the current index.
	DefaultContentID (see page 87)	Gets or Sets the Default ContentID for this Control.
	Direction (see page 88)	Direction for sorting results, "ascending" or "descending".
	DynamicParameter (see page 88)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)
	EkItems (see page 88)	Returns the array of ContentBase.
	EnablePaging (see page 89)	Enables paging if total records exceeds MaxResult value
	GetHtml (see page 89)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	IncludeIcons (see page 89)	Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.
	IsSynchronized (see page 90)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 90)	Gets or Sets the target window for the result
	MaxResults (see page 90)	Maximum number of results to be returned.
	MetaTag (see page 90)	Gets or Sets the meta tag value. Once this value is set, associated content (see page 337)(s) will be listed.
	OrderKey (see page 91)	Sort order key.
	Random (see page 91)	If set to true, grabs a single content (see page 337) block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the collection.
	SyncRoot (see page 91)	Gets an object that can be used to synchronize the ContentList.
	XmlDoc (see page 91)	Returns the Xml object loaded for ContentList.

Legend

	Method
	Property
V	virtual

ContentList Methods

	Name	Description
	CopyTo (see page 84)	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 84)	Populates the data for the ContentList.
	GetEnumerator (see page 84)	Gets a System.Collections.IEnumerator for the ContentList.
	GetRssFeed (see page 84)	This method returns the ContentList RSS feed.
	MoveNext (see page 85)	Move to next index.
	Reset (see page 85)	Resets the index.
	this (see page 85)	Returns the current ContentItem.ContentRow.
	ToString (see page 85)	Overrides the system ToString function.
	Wrap (see page 86)	Wrap the Html toolbar.

Legend

	Method

ContentList Properties

	Name	Description
	CacheInterval (see page 86)	Sets the cache interval in second(s).
	ContentIds (see page 87)	The comma separated list of content (see page 337) IDs you wish to display.
	Count (see page 87)	Indicates the number of elements in your ContentList.
	Current (see page 87)	Gets the current index.
	DefaultContentID (see page 87)	Gets or Sets the Default ContentID for this Control.
	Direction (see page 88)	Direction for sorting results, "ascending" or "descending".
	DynamicParameter (see page 88)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)
	EkItems (see page 88)	Returns the array of ContentBase.
	EnablePaging (see page 89)	Enables paging if total records exceeds MaxResult value
	GetHtml (see page 89)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	IncludeIcons (see page 89)	Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.
	IsSynchronized (see page 90)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 90)	Gets or Sets the target window for the result
	MaxResults (see page 90)	Maximum number of results to be returned.
	MetaTag (see page 90)	Gets or Sets the meta tag value. Once this value is set, associated content (see page 337)(s) will be listed.
	OrderKey (see page 91)	Sort order key.
	Random (see page 91)	If set to true, grabs a single content (see page 337) block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the collection.
	SyncRoot (see page 91)	Gets an object that can be used to synchronize the ContentList.
	XmlDoc (see page 91)	Returns the Xml object loaded for ContentList.

Legend

	Property
	virtual

ContentList Methods

The methods of the ContentList class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 84)	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 84)	Populates the data for the ContentList (see page 81).
	GetEnumerator (see page 84)	Gets a System.Collections.IEnumerator for the ContentList (see page 81).
	GetRssFeed (see page 84)	This method returns the ContentList (see page 81) RSS feed.
	MoveNext (see page 85)	Move to next index.
	Reset (see page 85)	Resets the index.
	this (see page 85)	Returns the current ContentItem.ContentRow.
	ToString (see page 85)	Overrides the system ToString function.
	Wrap (see page 86)	Wrap the Html toolbar.

Legend

	Method

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 81).

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 81).

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 81) 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
i	Index value

Returns

Content (see page 337) 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
Html	

ContentList Properties

The properties of the ContentList class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 86)	Sets the cache interval in second(s).
	ContentIds (see page 87)	The comma separated list of content (see page 337) IDs you wish to display.
	Count (see page 87)	Indicates the number of elements in your ContentList (see page 81).
	Current (see page 87)	Gets the current index.
	DefaultContentID (see page 87)	Gets or Sets the Default ContentID for this Control.
	Direction (see page 88)	Direction for sorting results, "ascending" or "descending".
	DynamicParameter (see page 88)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)
	EkItems (see page 88)	Returns the array of ContentBase.
	EnablePaging (see page 89)	Enables paging if total records exceeds MaxResult value
	GetHtml (see page 89)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	IncludeIcons (see page 89)	Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.
	IsSynchronized (see page 90)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 90)	Gets or Sets the target window for the result
	MaxResults (see page 90)	Maximum number of results to be returned.
	MetaTag (see page 90)	Gets or Sets the meta tag value. Once this value is set, associated content (see page 337)(s) will be listed.
	OrderKey (see page 91)	Sort order key.
	Random (see page 91)	If set to true, grabs a single content (see page 337) block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the collection.
	SyncRoot (see page 91)	Gets an object that can be used to synchronize the ContentList (see page 81).
	XmlDoc (see page 91)	Returns the Xml object loaded for ContentList (see page 81).

Legend

	Property
	virtual

ContentList.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)]
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
```

ContentList.ContentIds Property

The comma separated list of content (see page 337) 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()]
public string ContentIds;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The comma separated list of content IDs you wish to display.")>
<Personalizable()>
Public Property ContentIds() As string
```

ContentList.Count Property

Indicates the number of elements in your ContentList (see page 81).

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()]
public int DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Gets or Sets the Default ContentID for this Control.")>
<Personalizable()>
Public Property DefaultContentID() As Integer
```

ContentList.Direction Property

Direction for sorting results, "ascending" or "descending".

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("\\")]
[Description("The direction (ascending or descending) that you wish your sorted results to be
displayed in.")]
public Ektron.Cms.Controls.CmsWebService.OrderByDirection Direction;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("\\")>
<Description("The direction (ascending or descending) that you wish your sorted results to be
displayed in.")>
Public Property Direction() As Ektron.Cms.Controls.CmsWebService.OrderByDirection
```

ContentList.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 337) 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.EnablePaging Property

Enables paging if 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 MaxNumbers.")]
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 MaxNumbers.")>
Public Property EnablePaging() As bool
```

ContentList.GetHtml Property

If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) 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.")]
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.")>
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

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
```

ContentList.MaxResults Property

Maximum number of results to be returned.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The maximum number of results to fetch. If set to zero, there is no limit.")]
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.")>
Public Property MaxResults() As Integer
```

ContentList.MetaTag Property

Gets or Sets the meta tag value. Once this value is set, associated content (see page 337)(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.OrderKey Property

Sort order key.

C#

```
[Bindable(true)]
[DefaultValue(Ektron.Cms.Controls.CmsWebService.ContentListOrderBy.OrderOfTheIds)]
[Description("The key you wish to sort your results by.")]
[Category("Ektron")]
public Ektron.Cms.Controls.CmsWebService.ContentListOrderBy OrderKey;
```

Visual Basic

```
<Bindable(true)>
<DefaultValue(Ektron.Cms.Controls.CmsWebService.ContentListOrderBy.OrderOfTheIds)>
<Description("The key you wish to sort your results by.")>
<Category("Ektron")>
Public Property OrderKey() As Ektron.Cms.Controls.CmsWebService.ContentListOrderBy
```

ContentList.Random Property

If set to true, grabs a single content (see page 337) block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the collection.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Set this property to True to grab a single, random piece of content from this
collection.")]
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
collection.")>
Public Property Random() As bool
```

ContentList.SyncRoot Property

Gets an object that can be used to synchronize the ContentList (see page 81).

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 81).

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

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

ContentRating Class

The ContentRating server control.

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

Members

ContentRating Methods

	Name	Description
	Fill (see page 94)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 94)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 94)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
ContentRating Methods (see page 94)	The methods of the ContentRating class are listed here.
ContentRating Properties (see page 95)	The properties of the ContentRating class are listed here.

ContentRating Properties

	Name	Description
	AlreadyRatedMessage (see page 95)	The label displayed to a user thanking them for their content (see page 337) rating. Defaults to "Thank you for your feedback".
	BadLabel (see page 96)	The label corresponding to a "Bad" content (see page 337) rating. This defaults to "Poor".
	CharsRemainingLabel (see page 96)	
	CompleteGraphDisplay (see page 96)	Described the layout manner which the graph is displayed to a user after content (see page 337) 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 97)	
	DefaultContentID (see page 97)	Gets or Sets the Default ContentID for this Control. This is the ContentID that will be displayed if the DynamicParameter (see page 97) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 97) of "id", and will display DefaultContentID otherwise.)
	DynamicParameter (see page 97)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically. Set to "" (empty) if you wish to always display the default ContentID (static.)
	FeedbackHeader (see page 98)	The label corresponding to how the user would rate the content (see page 337) block. Defaults to "How would you rate the usefulness of this content (see page 337)?".
	GoodLabel (see page 98)	The label corresponding to a "Good" content (see page 337) rating. This defaults to "Outstanding".

	InitialGraphDisplay (see page 98)	Describes the layout manner which the graph is displayed to a user before content (see page 337) 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 99)	
	RatingLevelLabel (see page 99)	The label corresponding to a "Good" content (see page 337) rating. This defaults to "Outstanding".
	ThankYouMessage (see page 99)	The label displayed to a user thanking them for their content (see page 337) rating. Defaults to "Thank you for your feedback.".
	TotalRatingsLabel (see page 100)	
	UserCommentsHeader (see page 100)	The label corresponding to why the user rated content (see page 337) the way they did. Defaults to "Tell us why you rated the content (see page 337) this way".
	ValidationMessage (see page 100)	The label displayed to a user when they try to submit a content (see page 337) rating without selecting a value of 1-9. Defaults to "Please select a rating before submitting your feedback.".

Legend

	Method
	Property

ContentRating Methods

	Name	Description
	Fill (see page 94)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 94)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 94)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

ContentRating Properties

	Name	Description
	AlreadyRatedMessage (see page 95)	The label displayed to a user thanking them for their content (see page 337) rating. Defaults to "Thank you for your feedback".
	BadLabel (see page 96)	The label corresponding to a "Bad" content (see page 337) rating. This defaults to "Poor".
	CharsRemainingLabel (see page 96)	
	CompleteGraphDisplay (see page 96)	Described the layout manner which the graph is displayed to a user after content (see page 337) 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 97)	
	DefaultContentID (see page 97)	Gets or Sets the Default ContentID for this Control. This is the ContentID that will be displayed if the DynamicParameter (see page 97) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 97) of "id", and will display DefaultContentID otherwise.)
	DynamicParameter (see page 97)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically. Set to "" (empty) if you wish to always display the default ContentID (static.)
	FeedbackHeader (see page 98)	The label corresponding to how the user would rate the content (see page 337) block. Defaults to "How would you rate the usefulness of this content (see page 337)?".
	GoodLabel (see page 98)	The label corresponding to a "Good" content (see page 337) rating. This defaults to "Outstanding".
	InitialGraphDisplay (see page 98)	Describes the layout manner which the graph is displayed to a user before content (see page 337) 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 99)	
	RatingLevelLabel (see page 99)	The label corresponding to a "Good" content (see page 337) rating. This defaults to "Outstanding".
	ThankYouMessage (see page 99)	The label displayed to a user thanking them for their content (see page 337) rating. Defaults to "Thank you for your feedback.".
	TotalRatingsLabel (see page 100)	
	UserCommentsHeader (see page 100)	The label corresponding to why the user rated content (see page 337) the way they did. Defaults to "Tell us why you rated the content (see page 337) this way".

	ValidationMessage (see page 100)	The label displayed to a user when they try to submit a content (see page 337) rating without selecting a value of 1-9. Defaults to "Please select a rating before submitting your feedback.".

Legend

	Property

ContentRating Methods

The methods of the ContentRating class are listed here.

Public Methods

	Name	Description
	Fill (see page 94)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 94)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.
	Wrap (see page 94)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
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 95)	The label displayed to a user thanking them for their content (see page 337) rating. Defaults to "Thank you for your feedback".
	BadLabel (see page 96)	The label corresponding to a "Bad" content (see page 337) rating. This defaults to "Poor".
	CharsRemainingLabel (see page 96)	
	CompleteGraphDisplay (see page 96)	Described the layout manner which the graph is displayed to a user after content (see page 337) 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 97)	
	DefaultContentID (see page 97)	Gets or Sets the Default ContentID for this Control. This is the ContentID that will be displayed if the DynamicParameter (see page 97) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 97) of "id", and will display DefaultContentID otherwise.)
	DynamicParameter (see page 97)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically. Set to "" (empty) if you wish to always display the default ContentID (static.)
	FeedbackHeader (see page 98)	The label corresponding to how the user would rate the content (see page 337) block. Defaults to "How would you rate the usefulness of this content (see page 337)?".
	GoodLabel (see page 98)	The label corresponding to a "Good" content (see page 337) rating. This defaults to "Outstanding".
	InitialGraphDisplay (see page 98)	Describes the layout manner which the graph is displayed to a user before content (see page 337) 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 99)	
	RatingLevelLabel (see page 99)	The label corresponding to a "Good" content (see page 337) rating. This defaults to "Outstanding".
	ThankYouMessage (see page 99)	The label displayed to a user thanking them for their content (see page 337) rating. Defaults to "Thank you for your feedback.".
	TotalRatingsLabel (see page 100)	
	UserCommentsHeader (see page 100)	The label corresponding to why the user rated content (see page 337) the way they did. Defaults to "Tell us why you rated the content (see page 337) this way".
	ValidationMessage (see page 100)	The label displayed to a user when they try to submit a content (see page 337) rating without selecting a value of 1-9. Defaults to "Please select a rating before submitting your feedback.".

Legend

	Property

ContentRating.AlreadyRatedMessage Property

The label displayed to a user thanking them for their content (see page 337) 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 (see page 337) 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

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("TODO")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_CHARS)]
public string CharsRemainingLabel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("TODO")>
<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 (see page 337) 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

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("TODO")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_COOK)]
public string CookiesRequiredMessage;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("TODO")>
<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 97) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 97) 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 int 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 Integer
```

ContentRating.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 337) 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 (see page 337) block. Defaults to "How would you rate the usefulness of this content (see page 337)?".

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 (see page 337) 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.InitialGraphDisplay Property

Describes the layout manner which the graph is displayed to a user before content (see page 337) 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

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("TODO")]
public int Padding;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("TODO")>
Public Property Padding() As Integer
```

ContentRating.RatingLevelLabel Property

The label corresponding to a "Good" content (see page 337) 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 (see page 337) 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

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("TODO")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_TOTAL)]
public string TotalRatingsLabel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("TODO")>
<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 (see page 337) the way they did. Defaults to "Tell us why you rated the content (see page 337) 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 (see page 337) 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
```

DesignTimeDiagnostic Class

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

Description

This is class Ektron.Cms.Controls.DesignTimeDiagnostic.

Members

DesignTimeDiagnostic Methods

	Name	Description
	Fill (see page 102)	Method that can be overridden and used to grab data from the back end for each server control.

Topics

Name	Description
DesignTimeDiagnostic Methods (see page 102)	The methods of the DesignTimeDiagnostic class are listed here.
DesignTimeDiagnostic Properties (see page 102)	The properties of the DesignTimeDiagnostic class are listed here.

DesignTimeDiagnostic Properties

	Name	Description
	Authenticated (see page 103)	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 103)	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 103)	to be changed
	SuppressWrapperTags (see page 103)	Suppresses the output of span/div tags around the control.
	WrapTag (see page 104)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Method
	Property

DesignTimeDiagnostic Methods

	Name	Description
	Fill (see page 102)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

DesignTimeDiagnostic Properties

	Name	Description
	Authenticated (see page 103)	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 103)	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 103)	to be changed
	SuppressWrapperTags (see page 103)	Suppresses the output of span/div tags around the control.
	WrapTag (see page 104)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property

DesignTimeDiagnostic Methods

The methods of the DesignTimeDiagnostic class are listed here.

Public Methods

	Name	Description
	Fill (see page 102)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

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 103)	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 103)	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 103)	to be changed

	SuppressWrapperTags (see page 103)	Suppresses the output of span/div tags around the control.
	WrapTag (see page 104)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property

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

C#

```
[Browsable(false)]
public new int Language;
```

Visual Basic

```
<Browsable(false)>
Public new Property Language() As Integer
```

Description

to be changed

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

Summary description for DhtmlMenu.

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

Members

Public Methods

	Name	Description
	DhtmlMenu (see page 106)	Class constructor calls the base class's constructor

DhtmlMenu Methods

	Name	Description
	Fill (see page 106)	Calls up the database and updates the contents of the Menu if the data is out of sync with the specified parameters.
	Wrap (see page 107)	Method that can be overridden and used to wrap an Ektron Toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
DhtmlMenu Methods (see page 106)	The methods of the DhtmlMenu class are listed here.
DhtmlMenu Properties (see page 107)	The properties of the DhtmlMenu class are listed here.

DhtmlMenu Properties

	Name	Description
	AccessKey (see page 108)	Ignored. The property has no effect.
	Align (see page 108)	Specify the horizontal alignment for the root menu text.
	BackColor (see page 108)	BackColor sets the menu background color.
	BorderColor (see page 108)	Sets the border color around the editor.
	BorderStyle (see page 109)	Sets the border style around the editor.
	BorderWidth (see page 109)	Sets the border width around the editor.

	CssClass (see page 109)	Specify the class name to use instead of internal style.
	DisplayXslt (see page 109)	Ignored. The property has no effect.
	Enabled (see page 109)	Ignored. The property has no effect.
	EnableIE6FormsFix (see page 110)	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 110)	The height (in pixels or percent) of the editor, including toolbars.
	HorizontalDisplay (see page 110)	Popup root menu horizontally.
	MenuHoverOverColor (see page 110)	`HoverOverColor` sets the menu hoverover color.
	MenuStyle (see page 111)	Specify the custom style to use. If specified, all specified appearance properties will be ignored.
	Stylesheet (see page 111)	Specify the style sheet relative to site root. Leave blank for default.
	SubMenuBackColor (see page 111)	`SubMenuBackColor` sets the sub menu background color.
	SubMenuHoverOverColor (see page 112)	`SubMenuHoverOverColor` sets the menu hoverover color.
	SubMenuItemHeight (see page 112)	`SubMenuItemHeight` is the height in pixels of submenu items.
	SubMenuItemWidth (see page 112)	`SubMenuItemWidth` is the width in pixels of submenu items.
	SubMenuTopOffset (see page 113)	`SubMenuTopOffset` sets the horizontal offset of submenu popups relative to the parent submenu item.
	TabIndex (see page 113)	Ignored. The property has no effect.
	ToolTip (see page 113)	Ignored. The property has no effect.
	Width (see page 113)	The width (in pixels or percent) of the editor, including toolbars.

Legend

	Constructor
	Property

DhtmlMenu Methods

	Name	Description
	Fill (see page 106)	Calls up the database and updates the contents of the Menu if the data is out of sync with the specified parameters.
	Wrap (see page 107)	Method that can be overridden and used to wrap an Ektron Toolbar corresponding to a particular Control around some Html text.

Legend

	Method

DhtmlMenu Properties

	Name	Description
	AccessKey (see page 108)	Ignored. The property has no effect.
	Align (see page 108)	Specify the horizontal alignment for the root menu text.
	BackColor (see page 108)	BackColor sets the menu background color.
	BorderColor (see page 108)	Sets the border color around the editor.
	BorderStyle (see page 109)	Sets the border style around the editor.
	BorderWidth (see page 109)	Sets the border width around the editor.
	CssClass (see page 109)	Specify the class name to use instead of internal style.
	DisplayXslt (see page 109)	Ignored. The property has no effect.
	Enabled (see page 109)	Ignored. The property has no effect.
	EnableIE6FormsFix (see page 110)	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 110)	The height (in pixels or percent) of the editor, including toolbars.
	HorizontalDisplay (see page 110)	Popup root menu horizontally.

	MenuHoverOverColor (see page 110)	`HoverOverColor` sets the menu hoverover color.
	MenuStyle (see page 111)	Specify the custom style to use. If specified, all specified appearance properties will be ignored.
	Stylesheet (see page 111)	Specify the style sheet relative to site root. Leave blank for default.
	SubMenuBackColor (see page 111)	`SubMenuBackColor` sets the sub menu background color.
	SubMenuHoverOverColor (see page 112)	`SubMenuHoverOverColor` sets the menu hoverover color.
	SubMenuItemHeight (see page 112)	`SubMenuItemHeight` is the height in pixels of submenu items.
	SubMenuItemWidth (see page 112)	`SubMenuItemWidth` is the width in pixels of submenu items.
	SubMenuTopOffset (see page 113)	`SubMenuTopOffset` sets the horizontal offset of submenu popups relative to the parent submenu item.
	TabIndex (see page 113)	Ignored. The property has no effect.
	ToolTip (see page 113)	Ignored. The property has no effect.
	Width (see page 113)	The width (in pixels or percent) of the editor, including toolbars.

Legend

	Property

Public Methods

	Name	Description
	DhtmlMenu (see page 106)	Class constructor calls the base class's constructor

Legend

	Constructor

DhtmlMenu.DhtmlMenu Constructor

C#

```
public DhtmlMenu();
```

Visual Basic

```
Public Sub DhtmlMenu()
```

Description

Class constructor calls the base class's constructor

DhtmlMenu Methods

The methods of the DhtmlMenu class are listed here.

Public Methods

	Name	Description
	Fill (see page 106)	Calls up the database and updates the contents of the Menu if the data is out of sync with the specified parameters.
	Wrap (see page 107)	Method that can be overridden and used to wrap an Ektron Toolbar corresponding to a particular Control around some Html text.

Legend

	Method

DhtmlMenu.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
```

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
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 108)	Ignored. The property has no effect.
	Align (see page 108)	Specify the horizontal alignment for the root menu text.
	BackColor (see page 108)	BackColor sets the menu background color.
	BorderColor (see page 108)	Sets the border color around the editor.
	BorderStyle (see page 109)	Sets the border style around the editor.
	BorderWidth (see page 109)	Sets the border width around the editor.
	CssClass (see page 109)	Specify the class name to use instead of internal style.
	DisplayXslt (see page 109)	Ignored. The property has no effect.
	Enabled (see page 109)	Ignored. The property has no effect.
	EnableIE6FormsFix (see page 110)	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 110)	The height (in pixels or percent) of the editor, including toolbars.
	HorizontalDisplay (see page 110)	Popup root menu horizontally.
	MenuHoverOverColor (see page 110)	HoverOverColor sets the menu hoverover color.
	MenuStyle (see page 111)	Specify the custom style to use. If specified, all specified appearance properties will be ignored.
	Stylesheet (see page 111)	Specify the style sheet relative to site root. Leave blank for default.
	SubMenuBackColor (see page 111)	SubMenuBackColor sets the sub menu background color.
	SubMenuHoverOverColor (see page 112)	SubMenuHoverOverColor sets the menu hoverover color.
	SubMenuItemHeight (see page 112)	SubMenuItemHeight is the height in pixels of submenu items.
	SubMenuItemWidth (see page 112)	SubMenuItemWidth is the width in pixels of submenu items.
	SubMenuTopOffset (see page 113)	SubMenuTopOffset sets the horizontal offset of submenu popups relative to the parent submenu item.

	TabIndex (see page 113)	Ignored. The property has no effect.
	ToolTip (see page 113)	Ignored. The property has no effect.
	Width (see page 113)	The width (in pixels or percent) of the editor, including toolbars.

Legend

	Property

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.

C#

```
[Description("Specify the horizontal alignment for the root menu text."), Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(HorizontalAlign))]
public HorizontalAlign Align;
```

Visual Basic

```
<Description("Specify the horizontal alignment for the root menu text."), 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

Specify the class name to use instead of internal style.

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
```

EktronCatalogPart Class

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

Description

This is class Ektron.Cms.Controls.EktronCatalogPart.

Members

EktronCatalogPart Properties

	Name	Description
	DefaultContentID (see page 115)	to be changed

Topics

Name	Description
EktronCatalogPart Properties (see page 114)	The properties of the EktronCatalogPart class are listed here.

Legend

	Property

EktronCatalogPart Properties

	Name	Description
	DefaultContentID (see page 115)	to be changed

Legend

	Property

EktronCatalogPart Properties

The properties of the EktronCatalogPart class are listed here.

Public Properties

	Name	Description
	DefaultContentID (see page 115)	to be changed

Legend

	Property

EktronCatalogPart.DefaultContentID Property

C#

```
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Browsable(true), DefaultValue(0)]
[Category("Ektron Content Block")]
public int DefaultContentID;
```

Visual Basic

```
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron Content Block")>
Public Property DefaultContentID() As Integer
```

Description

to be changed

EktronCommunityCatalogPart Class

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

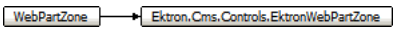
CommunityCatalogPart.cs

Description

This is class Ektron.Cms.Controls.EktronCommunityCatalogPart.

EktronWebPartZone Class

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

Description

This is class Ektron.Cms.Controls.EktronWebPartZone.

Members

EktronWebPartZone Properties

	Name	Description
	CmsSitePath (see page 116)	to be changed

Topics

Name	Description
EktronWebPartZone Properties (see page 116)	The properties of the EktronWebPartZone class are listed here.

Legend

	Property

EktronWebPartZone Properties

	Name	Description
	CmsSitePath (see page 116)	to be changed

Legend

	Property

EktronWebPartZone Properties

The properties of the EktronWebPartZone class are listed here.

Public Properties

	Name	Description
	CmsSitePath (see page 116)	to be changed

Legend

	Property

EktronWebPartZone.CmsSitePath Property

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
```

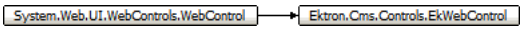
Description

to be changed

EkWebControl Class

The base class that all Ektron CMS400 Server Controls (see page 5) derive from

Class Hierarchy



C#

```
[ToolboxData("<{0}:EkWebControl runat=server></{0}:EkWebControl>")]
public abstract class EkWebControl : System.Web.UI.WebControls.WebControl;
```

Visual Basic

```
<ToolboxData("<{0}:EkWebControl runat=server></{0}:EkWebControl>")>
Public abstract Class EkWebControl
Inherits System.Web.UI.WebControls.WebControl
```

File

EkWebControl.cs

Members

Public Methods

	Name	Description
	EkWebControl (see page 119)	Constructor for the Ektron Server Control base class - initializes the CmsConnection layer

EkWebControl Methods

	Name	Description
	Fill (see page 119)	Method that can be overridden and used to grab data from the back end for each server control.
	LocalizeString (see page 120)	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 class.
	RegisterClientScriptInclude (see page 120)	This method adds JavaScript include in the page header. You can call this method multiple times with same script path because it will only register once.
	RegisterStylesheet (see page 120)	This methods adds a stylesheet link tag in the page header. You can call this method multiple times with same stylesheet path because it will only register once.
	SeachHeader (see page 120)	This methods searches for the literal-control containing the head-definition
	ToString (see page 121)	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 121)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
EkWebControl Methods (see page 119)	The methods of the EkWebControl class are listed here.
EkWebControl Properties (see page 121)	The properties of the EkWebControl class are listed here.

EkWebControl Properties

	Name	Description
	Authenticated (see page 122)	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.

	Error (see page 122)	Gets or Sets the Default ContentID for this Control.
	Hide (see page 122)	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 123)	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 123)	to be changed
	LanguageID (see page 123)	This property lets a designer view content (see page 337) blocks, collections etc in different languages.
	loggedInUserID (see page 123)	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 124)	DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currently logged in.
	SuppressWrapperTags (see page 124)	Suppresses the output of span/div tags around the control.
	Text (see page 124)	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 124)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Constructor
	virtual
	Property

EkWebControl Methods

	Name	Description
	Fill (see page 119)	Method that can be overridden and used to grab data from the back end for each server control.
	LocalizeString (see page 120)	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 class.
	RegisterClientScriptInclude (see page 120)	This method adds JavaScript include in the page header. You can call this method multiple times with same script path because it will only register once.
	RegisterStylesheet (see page 120)	This methods adds a stylesheet link tag in the page header. You can call this method multiple times with same stylesheet path because it will only register once.
	SeachHeader (see page 120)	This methods searches for the literal-control containing the head-definition
	ToString (see page 121)	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 121)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method
	virtual

EkWebControl Properties

	Name	Description
	Authenticated (see page 122)	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.
	Error (see page 122)	Gets or Sets the Default ContentID for this Control.
	Hide (see page 122)	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 123)	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 123)	to be changed
	LanguageID (see page 123)	This property lets a designer view content (see page 337) blocks, collections etc in different languages.
	loggedInUserID (see page 123)	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 124)	DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currently logged in.
	SuppressWrapperTags (see page 124)	Suppresses the output of span/div tags around the control.

	Text (see page 124)	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 124)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property
	virtual

Public Methods

	Name	Description
	EkWebControl (see page 119)	Constructor for the Ektron Server Control base class - initializes the CmsConnection layer

Legend

	Constructor

EkWebControl.EkWebControl Constructor

Constructor for the Ektron Server Control base class - initializes the CmsConnection layer

C#

```
public EkWebControl();
```

Visual Basic

```
Public Sub EkWebControl()
```

EkWebControl Methods

The methods of the EkWebControl class are listed here.

Public Methods

	Name	Description
	Fill (see page 119)	Method that can be overridden and used to grab data from the back end for each server control.
	LocalizeString (see page 120)	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 117) class.
	RegisterClientScriptInclude (see page 120)	This method adds JavaScript include in the page header. You can call this method multiple times with same script path because it will only register once.
	RegisterStylesheet (see page 120)	This methods adds a stylesheet link tag in the page header. You can call this method multiple times with same stylesheet path because it will only register once.
	SeachHeader (see page 120)	This methods searches for the literal-control containing the head-definition
	ToString (see page 121)	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 121)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method
	virtual

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.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 117) class.

C#

```
public string LocalizeString(string strInputText);
```

Visual Basic

```
Public Function LocalizeString(strInputText As string) As string
```

Parameters

Parameters	Description
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.

EkWebControl.RegisterClientScriptInclude Method

This method adds JavaScript include in the page header. You can call this method multiple times with same script path because it will only register once.

C#

```
public void RegisterClientScriptInclude(string path);
```

Visual Basic

```
Public Function RegisterClientScriptInclude(path As string) As void
```

Parameters

Parameters	Description
path	JavaScript include path

EkWebControl.RegisterStylesheet Method

This methods adds a stylesheet link tag in the page header. You can call this method multiple times with same stylesheet path because it will only register once.

C#

```
public void RegisterStylesheet(string path);
```

Visual Basic

```
Public Function RegisterStylesheet(path As string) As void
```

Parameters

Parameters	Description
path	Stylesheet path

EkWebControl.SeachHeader Method

This methods searches for the literal-control containing the head-definition

C#

```
public LiteralControl SeachHeader(ref int index);
```

Visual Basic

```
Public Function SeachHeader(index As ref int) As LiteralControl
```

Parameters

Parameters	Description
index	is used to store the position of the head-definition in the literal-control

Returns

returns the LiteralControl containing the head-definition, if not found it is null

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 virtual string Wrap(string Html);
```

Visual Basic

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

Parameters

Parameters	Description
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 122)	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.
	Error (see page 122)	Gets or Sets the Default ContentID for this Control.
	Hide (see page 122)	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 123)	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 123)	to be changed
	LanguageID (see page 123)	This property lets a designer view content (see page 337) blocks, collections etc in different languages.
	loggedInUserID (see page 123)	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 124)	DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currently logged in.

	SuppressWrapperTags (see page 124)	Suppresses the output of span/div tags around the control.
	Text (see page 124)	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 124)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property
	virtual

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.Error Property

Gets or Sets the Default ContentID for this 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

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(1033)]
[Description("Property that lets a view content blocks, collections etc in different
languages.")]
[Editor(typeof(SelectLanguageIDTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int Language;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(1033)>
<Description("Property that lets a view content blocks, collections etc in different
languages.")>
<Editor(typeof(SelectLanguageIDTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property Language() As Integer
```

Description

to be changed

EkWebControl.LanguageID Property

This property lets a designer view content (see page 337) blocks, collections etc in different languages.

C#

```
public int LanguageID;
```

Visual Basic

```
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 int loggedInUserID;
```

Visual Basic

```
Public Property loggedInUserID() As Integer
```

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.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#

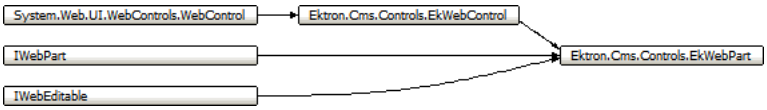
```
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("span")]
[TypeConverter(typeof(WrapTagTypeConverter))]
[Description("Allows a developer to specify control's tag. Default is assumed '<span>'.")]
public string WrapTag;
```

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 Property WrapTag() As string
```

EkWebPart Class

Class Hierarchy



C#

```
public abstract class EkWebPart : EkWebControl, IWebPart, IWebEditable;
```

Visual Basic

```
Public abstract Class EkWebPart
Inherits EkWebControl
Inherits IWebPart
Inherits IWebEditable
```

File

EkWebPart.cs

Description

This is class Ektron.Cms.Controls.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

Members

Public Methods

	Name	Description
	EkXsltBase (see page 126)	Constructor for EkXlstBase

EkXsltBase Methods

	Name	Description
	SetDirty (see page 127)	Marks the Control to call the database since output-dependent parameters have been changed.

Topics

Name	Description
EkXsltBase Methods (see page 127)	The methods of the EkXsltBase class are listed here.
EkXsltBase Properties (see page 127)	The properties of the EkXsltBase class are listed here.

EkXsltBase Properties

	Name	Description
	DisplayXslt (see page 127)	The Xslt to use to render the control (optional).
	XmlDoc (see page 128)	XmlDocument used in this control.
	XsltDoc (see page 128)	Xslt used in this control.

Legend

	Constructor
	Property
	virtual

EkXsltBase Methods

	Name	Description
	SetDirty (see page 127)	Marks the Control to call the database since output-dependent parameters have been changed.

Legend

	Method

EkXsltBase Properties

	Name	Description
	DisplayXslt (see page 127)	The Xslt to use to render the control (optional).
	XmlDoc (see page 128)	XmlDocument used in this control.
	XsltDoc (see page 128)	Xslt used in this control.

Legend

	Property
	virtual

Public Methods

	Name	Description
	EkXsltBase (see page 126)	Constructor for EkXlstBase

Legend

	Constructor

EkXsltBase.EkXsltBase Constructor

Constructor for EkXlstBase

C#

```
public EkXsltBase();
```

Visual Basic

```
Public Sub EkXsltBase()
```

EkXsltBase Methods

The methods of the EkXsltBase class are listed here.

Public Methods

	Name	Description
	SetDirty (see page 127)	Marks the Control to call the database since output-dependent parameters have been changed.

Legend

	Method

EkXsltBase.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
```

EkXsltBase Properties

The properties of the EkXsltBase class are listed here.

Public Properties

	Name	Description
	DisplayXslt (see page 127)	The Xslt to use to render the control (optional).
	XmlDoc (see page 128)	XmlDocument used in this control.
	XsltDoc (see page 128)	Xslt used in this control.

Legend

	Property
	virtual

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.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#

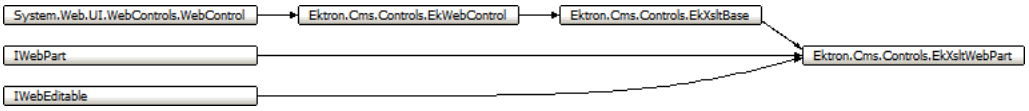
```
[Browsable(false)]
public virtual XmlDocument XsltDoc;
```

Visual Basic

```
<Browsable(false)>
Public virtual Property XsltDoc() As XmlDocument
```

EkXsltWebPart Class

Class Hierarchy



C#

```
public abstract class EkXsltWebPart : EkXsltBase, IWebPart, IWebEditable;
```

Visual Basic

```
Public abstract Class EkXsltWebPart
Inherits EkXsltBase
Inherits IWebPart
Inherits IWebEditable
```

File

EkXsltWebPart.cs

Description

This is class Ektron.Cms.Controls.EkXsltWebPart.

FolderBreadcrumb Class

This control display the folder path.

Class Hierarchy

C#

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

Visual Basic

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

File

FolderBreadcrumb.cs

Members

FolderBreadcrumb 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.

Topics

Name	Description
FolderBreadcrumb Methods (see page 129)	The methods of the FolderBreadcrumb class are listed here.
FolderBreadcrumb Properties (see page 130)	The properties of the FolderBreadcrumb class are listed here.

FolderBreadcrumb Properties

	Name	Description
	DynamicParameter (see page 130)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically - Set to String.Empty if you wish to always display the default ContentID (static.)

Legend

	Method
	Property

FolderBreadcrumb 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.

Legend

	Method

FolderBreadcrumb Properties

	Name	Description
	DynamicParameter (see page 130)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically - Set to String.Empty if you wish to always display the default ContentID (static.)

Legend

	Property

FolderBreadcrumb Methods

The methods of the FolderBreadcrumb 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.

Legend

	Method

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
	DynamicParameter (see page 130)	Gets or Sets the QueryString parameter to read a Content (see page 337) ID dynamically - Set to String.Empty if you wish to always display the default ContentID (static.)

Legend

	Property

FolderBreadcrumb.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 337) 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;
```

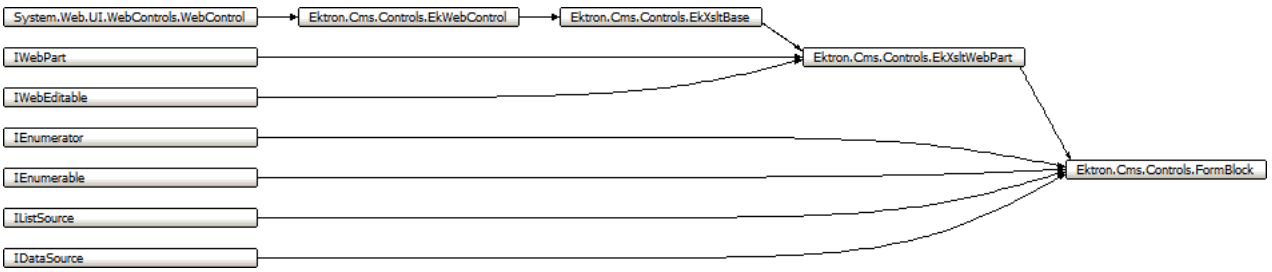
Visual Basic

```
<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
```

FormBlock Class

Summary description for FormBlock.

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

Members

FormBlock Methods

	Name	Description
	CopyTo (see page 133)	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 133)	Overrideable method that can be used to grab data from the back end for each server control.
	GetEnumerator (see page 133)	Gets a System.Collections.IEnumerator for the collection.
	GetView (see page 133)	Gets the view data using a dataset.
	GetViewNames (see page 134)	Gets the view names.
	MoveNext (see page 134)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 134)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 134)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	ToString (see page 135)	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 135)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
FormBlock Methods (see page 132)	The methods of the FormBlock class are listed here.
FormBlock Properties (see page 135)	The properties of the FormBlock class are listed here.

FormBlock Properties

	Name	Description
	AddValidation (see page 136)	Set this value to true if you wish to add the validation to your main runat=server form.
	CacheInterval (see page 136)	Sets the cache interval in second(s).
	Count (see page 136)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 136)	Gets the current object in the collection.
	DynamicParameter (see page 137)	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.)
	EkItem (see page 137)	Gets the Ektron.Cms.Common.ContentBase for the current Form Block.
	Fields (see page 137)	Returns a collection of fields defined in the selected form.
	FormName (see page 137)	Set this value to the name of your main runat=server form.
	IncludeTags (see page 138)	Set this to true if you wish to have this Ektron Form element to wrap its own Form tags around itself.
	IsSynchronized (see page 138)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

	StartingFolderID (see page 138)	WEB PARTS ONLY: Allows a developer to set the starting folder for the personalization/collaboration selection dialog.
	SyncRoot (see page 139)	Gets an object that can be used to synchronize the collection.

Legend

	Method
	Property
	virtual

FormBlock Methods

	Name	Description
	CopyTo (see page 133)	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 133)	Overrideable method that can be used to grab data from the back end for each server control.
	GetEnumerator (see page 133)	Gets a System.Collections.IEnumerator for the collection.
	GetView (see page 133)	Gets the view data using a dataset.
	GetViewNames (see page 134)	Gets the view names.
	MoveNext (see page 134)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 134)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 134)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	ToString (see page 135)	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 135)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

FormBlock Properties

	Name	Description
	AddValidation (see page 136)	Set this value to true if you wish to add the validation to your main runat=server form.
	CacheInterval (see page 136)	Sets the cache interval in second(s).
	Count (see page 136)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 136)	Gets the current object in the collection.
	DynamicParameter (see page 137)	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.)
	EkItem (see page 137)	Gets the Ektron.Cms.Common.ContentBase for the current Form Block.
	Fields (see page 137)	Returns a collection of fields defined in the selected form.
	FormName (see page 137)	Set this value to the name of your main runat=server form.
	IncludeTags (see page 138)	Set this to true if you wish to have this Ektron Form element to wrap its own Form tags around itself.
	IsSynchronized (see page 138)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	StartingFolderID (see page 138)	WEB PARTS ONLY: Allows a developer to set the starting folder for the personalization/collaboration selection dialog.
	SyncRoot (see page 139)	Gets an object that can be used to synchronize the collection.

Legend

	Property
	virtual

FormBlock Methods

The methods of the FormBlock class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 133)	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 133)	Overrideable method that can be used to grab data from the back end for each server control.
	GetEnumerator (see page 133)	Gets a System.Collections.IEnumerator for the collection.
	GetView (see page 133)	Gets the view data using a dataset.
	GetViewNames (see page 134)	Gets the view names.
	MoveNext (see page 134)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 134)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 134)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	ToString (see page 135)	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 135)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
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 63). 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 63) 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
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 136)	Set this value to true if you wish to add the validation to your main runat=server form.
V	CacheInterval (see page 136)	Sets the cache interval in second(s).
	Count (see page 136)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 136)	Gets the current object in the collection.
	DynamicParameter (see page 137)	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.)
	EkItem (see page 137)	Gets the Ektron.Cms.Common.ContentBase for the current Form Block.
	Fields (see page 137)	Returns a collection of fields defined in the selected form.
	FormName (see page 137)	Set this value to the name of your main runat=server form.
	IncludeTags (see page 138)	Set this to true if you wish to have this Ektron Form element to wrap its own Form tags around itself.
	IsSynchronized (see page 138)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	StartingFolderID (see page 138)	WEB PARTS ONLY: Allows a developer to set the starting folder for the personalization/collaboration selection dialog.
	SyncRoot (see page 139)	Gets an object that can be used to synchronize the collection.

Legend

	Property
V	virtual

FormBlock.AddValidation Property

Set this value to true if you wish to add the validation to your main runat=server form.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set this value to true if you wish to add the validation to your main
runat=server form.")]
[Personalizable()]
public bool AddValidation;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set this value to true if you wish to add the validation to your main
runat=server form.")>
<Personalizable()>
Public Property AddValidation() As bool
```

FormBlock.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()]
public virtual double CacheInterval;
```

Visual Basic

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

FormBlock.Count Property

Indicates the number of elements in your Collection (see page 63).

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.DynamicParameter Property

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.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[TypeConverter(typeof(DynamicFormIDTypeConverter))]
[Description("The QueryString parameter to retrieve a dynamic ID from.")]
[Personalizable()]
public string DynamicParameter;
```

Visual Basic

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

FormBlock.EkItem Property

Gets the Ektron.Cms.Common.ContentBase for the current Form 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 this to true if you wish to have this Ektron Form element to wrap its own Form tags around itself.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Whether or not to wrap form tags around this Ektron Form element.")]
[Personalizable()]
public bool IncludeTags;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Whether or not to wrap form tags around this Ektron Form element.")>
<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 int 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 Integer
```

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

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

Members

FormFieldItem Properties

	Name	Description
	Basetype (see page 140)	Gets the basic datatype of the field.
	ContentType (see page 140)	Gets the type of content (see page 337) allowed within the field. This is loosely associated with the datatype.
	DatalistItems (see page 141)	Gets the list of options for this field's value, if applicable, otherwise null/Nothing.
	Datatype (see page 141)	Gets the datatype of the field.
	DisplayName (see page 141)	Gets the name of the field to be displayed.
	FieldName (see page 142)	Gets the name of the field.
	Language (see page 142)	Gets the language code of the datalist. For example, en or en-US.
	Value (see page 142)	Gets or set the value of this field.

Topics

Name	Description
FormFieldItem Properties (see page 140)	The properties of the FormFieldItem class are listed here.

Legend

	Property

FormFieldItem Properties

	Name	Description
	Basetype (see page 140)	Gets the basic datatype of the field.
	ContentType (see page 140)	Gets the type of content (see page 337) allowed within the field. This is loosely associated with the datatype.
	DatalistItems (see page 141)	Gets the list of options for this field's value, if applicable, otherwise null/Nothing.
	Datatype (see page 141)	Gets the datatype of the field.
	DisplayName (see page 141)	Gets the name of the field to be displayed.
	FieldName (see page 142)	Gets the name of the field.
	Language (see page 142)	Gets the language code of the datalist. For example, en or en-US.
	Value (see page 142)	Gets or set the value of this field.

Legend

	Property

FormFieldItem Properties

The properties of the FormFieldItem class are listed here.

Public Properties

	Name	Description
	Basetype (see page 140)	Gets the basic datatype of the field.
	ContentType (see page 140)	Gets the type of content (see page 337) allowed within the field. This is loosely associated with the datatype.
	DatalistItems (see page 141)	Gets the list of options for this field's value, if applicable, otherwise null/Nothing.
	Datatype (see page 141)	Gets the datatype of the field.
	DisplayName (see page 141)	Gets the name of the field to be displayed.
	FieldName (see page 142)	Gets the name of the field.
	Language (see page 142)	Gets the language code of the datalist. For example, en or en-US.
	Value (see page 142)	Gets or set the value of this field.

Legend

	Property

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 (see page 336), and checkbox.

FormFieldItem.ContentType Property

Gets the type of content (see page 337) 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 (see page 337) 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#

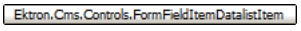
```
[Browsable(false)]
[Category("Data")]
[Description("The value of this field.")]
public string Value;
```

Visual Basic

```
<Browsable(false)>
<Category("Data")>
<Description("The value of this field.")>
Public Property Value() As string
```

FormFieldItemDatalistItem Class

Class Hierarchy



C#

```
[DefaultProperty("DataValue")]
```

```
[ValidationProperty("DataValue")]
[TypeConverter(typeof(FormFieldItemDatalistItemConverter))]
public class FormFieldItemDatalistItem;
```

Visual Basic

```
<DefaultProperty("DataValue")>
<ValidationProperty("DataValue")>
<TypeConverter(typeof(FormFieldItemDatalistItemConverter))>
Public Class FormFieldItemDatalistItem
```

File

FormField.cs

Description

This is class Ektron.Cms.Controls.FormFieldItemDatalistItem.

Members

FormFieldItemDatalistItem Properties

	Name	Description
	DataValue (see page 143)	Value suitable for XML or programmatic processing.
	DisplayValue (see page 144)	Value suitable for display.

Topics

Name	Description
FormFieldItemDatalistItem Properties (see page 143)	The properties of the FormFieldItemDatalistItem class are listed here.

Legend

	Property

FormFieldItemDatalistItem Properties

	Name	Description
	DataValue (see page 143)	Value suitable for XML or programmatic processing.
	DisplayValue (see page 144)	Value suitable for display.

Legend

	Property

FormFieldItemDatalistItem Properties

The properties of the FormFieldItemDatalistItem class are listed here.

Public Properties

	Name	Description
	DataValue (see page 143)	Value suitable for XML or programmatic processing.
	DisplayValue (see page 144)	Value suitable for display.

Legend

	Property

FormFieldItemDatalistItem.DataValue Property

Value suitable for XML or programmatic processing.

C#

```
[Category("Data")]
[Description("Value suitable for XML or programmatic processing.")]
```

```
public string DataValue;
```

Visual Basic

```
<Category("Data")>
<Description("Value suitable for XML or programmatic processing.")>
Public Property DataValue() As string
```

FormFieldItemDatalistItem.DisplayValue Property

Value suitable for display.

C#

```
[Category("Appearance")]
[Description("Value suitable for display.")]
public string DisplayValue;
```

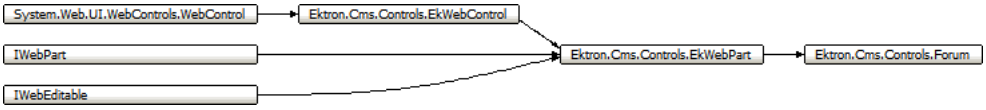
Visual Basic

```
<Category("Appearance")>
<Description("Value suitable for display.")>
Public Property DisplayValue() As string
```

Forum Class

Summary description for Forum.

Class Hierarchy



C#

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

Visual Basic

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

File

Forum.cs

Members

Forum Methods

	Name	Description
	Fill (see page 145)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 145)	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 146)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
Forum Methods (see page 145)	The methods of the Forum class are listed here.
Forum Properties (see page 146)	The properties of the Forum class are listed here.

Forum Properties

	Name	Description
	CacheInterval (see page 146)	sets the cache interval in second(s)

Legend

	Method
	Property
	virtual

Forum Methods

	Name	Description
	Fill (see page 145)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 145)	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 146)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

Forum Properties

	Name	Description
	CacheInterval (see page 146)	sets the cache interval in second(s)

Legend

	Property
	virtual

Forum Methods

The methods of the Forum class are listed here.

Public Methods

	Name	Description
	Fill (see page 145)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 145)	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 146)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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.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
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
	CacheInterval (see page 146)	sets the cache interval in second(s)

Legend

	Property
	virtual

Forum.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()]
public virtual double CacheInterval;
```

Visual Basic

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

HtmlEditDisplay Class

Class Hierarchy

C#

```
public class HtmlEditDisplay;
```

Visual Basic

```
Public Class HtmlEditDisplay
```

File

HtmlEditDisplay.cs

Description

This is class Ektron.Cms.Controls.HtmlEditDisplay.

Members

HtmlEditDisplay Methods

	Name	Description
	BaseJavaScript (see page 148)	Create the javascript that handles the onready event and other CMS specific setup functionality.
	BeginJavaScript (see page 149)	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 149)	This will end the surrounding of the JavaScript created for the CMS.
S	MakeFieldNameFromId (see page 149)	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 150)	Creates the fields used when the content (see page 337) 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 (see page 337) title [ControlName]_ContentId = The ID of the content (see page 337) [ControlName]_Language = The language of the content (see page 337) The content (see page 337) title caption is surrounded with a span that has this ID: [ControlName]_TitleLabel All of the information for all the content (see page 337) information fields (except the content (see page 337) 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 150)
	SelectedJavaScript (see page 150)	Renders any JavaScript based on options set by the developer through the properties.

Topics

Name	Description
HtmlEditDisplay Methods (see page 148)	The methods of the HtmlEditDisplay class are listed here.

Legend

	Method
S	static

HtmlEditDisplay Methods

	Name	Description
	BaseJavaScript (see page 148)	Create the javascript that handles the onready event and other CMS specific setup functionality.

	BeginJavaScript (see page 149)	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 149)	This will end the surrounding of the JavaScript created for the CMS.
S	MakeFieldNameFromId (see page 149)	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 150)	Creates the fields used when the content (see page 337) 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 (see page 337) title [ControlName]_ContentId = The ID of the content (see page 337) [ControlName]_Language = The language of the content (see page 337) The content (see page 337) title caption is surrounded with a span that has this ID: [ControlName]_TitleLabel All of the information for all the content (see page 337) information fields (except the content (see page 337) 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 150)
	SelectedJavaScript (see page 150)	Renders any JavaScript based on options set by the developer through the properties.

Legend

	Method
S	static

HtmlEditDisplay Methods

The methods of the HtmlEditDisplay class are listed here.

Public Methods

	Name	Description
	BaseJavaScript (see page 148)	Create the javascript that handles the onready event and other CMS specific setup functionality.
	BeginJavaScript (see page 149)	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 149)	This will end the surrounding of the JavaScript created for the CMS.
S	MakeFieldNameFromId (see page 149)	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 150)	Creates the fields used when the content (see page 337) 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 (see page 337) title [ControlName]_ContentId = The ID of the content (see page 337) [ControlName]_Language = The language of the content (see page 337) The content (see page 337) title caption is surrounded with a span that has this ID: [ControlName]_TitleLabel All of the information for all the content (see page 337) information fields (except the content (see page 337) 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 150)
	SelectedJavaScript (see page 150)	Renders any JavaScript based on options set by the developer through the properties.

Legend

	Method
S	static

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
strTranslationFile	Use m_strTranslationFile for this.
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(int iContentId, int iLanguageID);
```

Visual Basic

```
Public Function BeginJavaScript(iContentId As Integer, iLanguageID As Integer) As string
```

Parameters

Parameters	Description
iContentId	Use m_iContentId for this.
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
strBaseId	The name, or ID of the editor.

strSuffix	The descriptive piece of the field name.

HtmlEditDisplay.RenderCMSFields Method

Creates the fields used when the content (see page 337) 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 (see page 337) title [ControlName]_ContentId = The ID of the content (see page 337) [ControlName]_Language = The language of the content (see page 337)

The content (see page 337) title caption is surrounded with a span that has this ID:

[ControlName]_TitleLabel

All of the information for all the content (see page 337) information fields (except the content (see page 337) 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 (see page 337) for publication CheckIn - Check in the content (see page 337) Save - Save the content (see page 337) and redisplay in the editor UndoCheckout - Undo the checkout of the content (see page 337)

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
strContentTitle	The title value text.
strTitleLabel	Use RetrieveMessage(m_enumMessages.emsgContentTitle) for this.
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 (see page 337). The eWebEditPro editor can easily be placed onto a page for content (see page 337) 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 (see page 337) and can be processed in the receiving page.

[ControlName]_Action = the action to take - save, check in, undo checkout [ControlName]_ContentTitle = the content (see page 337) title [ControlName]_ContentId = the ID of the content (see page 337) [ControlName]_Language = the language of the content (see page 337)

The content (see page 337) title caption is surrounded with a span that has this ID:

[ControlName]_TitleLabel

All of the information for all the content (see page 337) information fields (except the content (see page 337) 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 proerties:

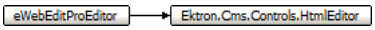
Field Name Read With [ControlName]_Action .ContentAction (see page 158) [ControlName]_ContentTitle .ContentTitle (see page 159) [ControlName]_ContentId .ContentId [ControlName]_Language .ContentLanguage [ControlName]_SafeEncode .SafeEncode (see page 163)

The content (see page 337) sent or posted is retrieved with the simple:

.Text

The valid values for the [ControlName]_Action value are: Publish - Submit the posted content (see page 337) for publication CheckIn - Check in the content (see page 337) Save - Save the content (see page 337) and redisplay in the editor UndoCheckout - Undo the checkout of the content (see page 337)

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

Members

Public Methods

	Name	Description
	HtmlEditor (see page 154)	Constructor of the content (see page 337) editor. There are no parameters. The control depends on the settings given at design time or at run time.

HtmlEditor Methods

	Name	Description
	ExecuteContentAction (see page 155)	This uses the ContentAction (see page 158) value and performs the specified action on the content (see page 337) contained within the server control's posted values. The content (see page 337) 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 155)	Returns the string which would be produced when the server control is rendered in the browser.
	WorkareaMode (see page 155)	Controls (see page 5) 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 (see page 337).

Topics

Name	Description
HtmlEditor Methods (see page 154)	The methods of the HtmlEditor class are listed here.

HtmlEditor Properties (see page 156)	The properties of the HtmlEditor class are listed here.

HtmlEditor Properties

	Name	Description
	Archived (see page 157)	This signifies that the selected content (see page 337) is archived.
	Authenticated (see page 157)	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 158)	DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time.
	ContentAction (see page 158)	The action to perform on the posted content (see page 337). 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.
	ContentTitle (see page 159)	The title for the selected Content (see page 337). This is set when new content (see page 337) is read. It can not be change 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 159)	This value describes the type of content (see page 337) the selected ID refers to. If this value is blank, then the content (see page 337) is a custom type, to be defined type, or unsupported type. These are the possible types: HTML Form
	EditorReadyCall (see page 160)	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 (see page 337). 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 160)	If this is 'true' then the user can upload locally selected images and linked files that they have put into their content (see page 337). If enabled, the automatic upload mechanism will allow upload the local files automatically with minimal user intervention. If this value is 'false' then the site developer must provide an upload page, such as the CMS library, where the upload can occur.
	IncludeContentInfo (see page 161)	If true, this will include the information about the content (see page 337) in fields posted with a form post. (This fields are defined in the summary of this control.) Set the value to 'false' if the page will maintain 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 164) property to false.
	IsLoggedIn (see page 161)	DESIGN TIME ONLY: Read only property that lets a developer know if the user is currently log into the site as a user.
	loggedInUserID (see page 162)	DESIGN TIME ONLY: Read only property that lets a developer know the user id of the user who is currenlty looged in.
	loggedInUserName (see page 162)	DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currenlty looged in.
	Path (see page 162)	The path to the site that contains this control. This is not available in the properties. 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 163)	The display to use for viewing the content (see page 337). This can be used to simplify the display or to debug the operation of editing content (see page 337) 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 163)	If true then the posted HTML is encoded using URL encoding. This makes the transfer secure. If the content (see page 337) is encoded then it must be decoded on the recieving 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 164)	Controls (see page 5) whether the CMS toolbar is displayed in the editor's toolbar. If this value is true, then commands containing the CMS functionality are displayed in the editor's toolbar. The commands are for check-in, publish, and edit cancellation.
	ShowTitle (see page 164)	If this is 'true' then the title of the selected content (see page 337) is displayed. The user can then edit the title. The submission of content (see page 337) posts up the title, even if it is not displayed to the user, unless IncludeContentInfo (see page 161) is set to a false value.
	StyleSheet (see page 165)	Assigns a single stylesheet.
	Text (see page 165)	The HTML text being edited.
	ToolbarLevel (see page 166)	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 57)", "FormDesign", "XSLTDesign", "Full", "Custom"

	ToolbarResetCall (see page 166)	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 enirely 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 167)	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.

Legend

	Constructor
	Property

HtmlEditor Methods

	Name	Description
	ExecuteContentAction (see page 155)	This uses the ContentAction (see page 158) value and performs the specified action on the content (see page 337) contained within the server control's posted values. The content (see page 337) 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 155)	Returns the string which would be produced when the server control is rendered in the browser.
	WorkareaMode (see page 155)	Controls (see page 5) 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 (see page 337).

Legend

	Method

HtmlEditor Properties

	Name	Description
	Archived (see page 157)	This signifies that the selected content (see page 337) is archived.
	Authenticated (see page 157)	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 158)	DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time.
	ContentAction (see page 158)	The action to perform on the posted content (see page 337). 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.
	ContentTitle (see page 159)	The title for the selected Content (see page 337). This is set when new content (see page 337) is read. It can not be change 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 159)	This value describes the type of content (see page 337) the selected ID refers to. If this value is blank, then the content (see page 337) is a custom type, to be defined type, or unsupported type. These are the possible types: HTML Form
	EditorReadyCall (see page 160)	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 (see page 337). 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 160)	If this is 'true' then the user can upload locally selected images and linked files that they have put into their content (see page 337). If enabled, the automatic upload mechanism will allow upload the local files automatically with minimal user intervention. If this value is 'false' then the site developer must provide an upload page, such as the CMS library, where the upload can occur.
	IncludeContentInfo (see page 161)	If true, this will include the information about the content (see page 337) in fields posted with a form post. (This fields are defined in the summary of this control.) Set the value to 'false' if the page will maintain 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 164) property to false.
	IsLoggedIn (see page 161)	DESIGN TIME ONLY: Read only property that lets a developer know if the user is currently log into the site as a user.
	loggedInUserID (see page 162)	DESIGN TIME ONLY: Read only property that lets a developer know the user id of the user who is currenlty looged in.
	loggedInUserName (see page 162)	DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currenlty looged in.

	Path (see page 162)	The path to the site that contains this control. This is not available in the properties. 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 163)	The display to use for viewing the content (see page 337). This can be used to simplify the display or to debug the operation of editing content (see page 337) 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 163)	If true then the posted HTML is encoded using URL encoding. This makes the transfer secure. If the content (see page 337) is encoded then it must be decoded on the recieving 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 164)	Controls (see page 5) whether the CMS toolbar is displayed in the editor's toolbar. If this value is true, then commands containing the CMS functionality are displayed in the editor's toolbar. The commands are for check-in, publish, and edit cancellation.
	ShowTitle (see page 164)	If this is 'true' then the title of the selected content (see page 337) is displayed. The user can then edit the title. The submission of content (see page 337) posts up the title, even if it is not displayed to the user, unless IncludeContentInfo (see page 161) is set to a false value.
	StyleSheet (see page 165)	Assigns a single stylesheet.
	Text (see page 165)	The HTML text being edited.
	ToolbarLevel (see page 166)	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 57)", "FormDesign", "XSLTDesign", "Full", "Custom"
	ToolbarResetCall (see page 166)	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 enirely 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 167)	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.

Legend

	Property

Public Methods

	Name	Description
	HtmlEditor (see page 154)	Constructor of the content (see page 337) editor. There are no parameters. The control depends on the settings given at design time or at run time.

Legend

	Constructor

HtmlEditor.HtmlEditor Constructor

Constructor of the content (see page 337) 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 HtmlEditor()
```

HtmlEditor Methods

The methods of the HtmlEditor class are listed here.

Public Methods

	Name	Description
	ExecuteContentAction (see page 155)	This uses the ContentAction (see page 158) value and performs the specified action on the content (see page 337) contained within the server control's posted values. The content (see page 337) 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 155)	Returns the string which would be produced when the server control is rendered in the browser.
	WorkareaMode (see page 155)	Controls (see page 5) 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 (see page 337).

Legend

	Method

HtmlEditor.ExecuteContentAction Method

This uses the ContentAction (see page 158) value and performs the specified action on the content (see page 337) contained within the server control's posted values. The content (see page 337) 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
```

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 5) 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 (see page 337).

C#

```
public void WorkareaMode(int iMode);
```

Visual Basic

```
Public Function WorkareaMode(iMode As Integer) As void
```

Parameters

Parameters	Description
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 157)	This signifies that the selected content (see page 337) is archived.
	Authenticated (see page 157)	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 158)	DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time.
	ContentAction (see page 158)	The action to perform on the posted content (see page 337). 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.
	ContentTitle (see page 159)	The title for the selected Content (see page 337). This is set when new content (see page 337) is read. It can not be change 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 159)	This value describes the type of content (see page 337) the selected ID refers to. If this value is blank, then the content (see page 337) is a custom type, to be defined type, or unsupported type. These are the possible types: HTML Form
	EditorReadyCall (see page 160)	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 (see page 337). 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 160)	If this is 'true' then the user can upload locally selected images and linked files that they have put into their content (see page 337). If enabled, the automatic upload mechanism will allow upload the local files automatically with minimal user intervention. If this value is 'false' then the site developer must provide an upload page, such as the CMS library, where the upload can occur.
	IncludeContentInfo (see page 161)	If true, this will include the information about the content (see page 337) in fields posted with a form post. (This fields are defined in the summary of this control.) Set the value to 'false' if the page will maintain 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 164) property to false.
	IsLoggedIn (see page 161)	DESIGN TIME ONLY: Read only property that lets a developer know if the user is currently log into the site as a user.
	loggedInUserID (see page 162)	DESIGN TIME ONLY: Read only property that lets a developer know the user id of the user who is currenlty looged in.
	loggedInUserName (see page 162)	DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currenlty looged in.
	Path (see page 162)	The path to the site that contains this control. This is not available in the properties. 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 163)	The display to use for viewing the content (see page 337). This can be used to simplify the display or to debug the operation of editing content (see page 337) 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 163)	If true then the posted HTML is encoded using URL encoding. This makes the transfer secure. If the content (see page 337) is encoded then it must be decoded on the recieving 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 164)	Controls (see page 5) whether the CMS toolbar is displayed in the editor's toolbar. If this value is true, then commands containing the CMS functionality are displayed in the editor's toolbar. The commands are for check-in, publish, and edit cancellation.
	ShowTitle (see page 164)	If this is 'true' then the title of the selected content (see page 337) is displayed. The user can then edit the title. The submission of content (see page 337) posts up the title, even if it is not displayed to the user, unless IncludeContentInfo (see page 161) is set to a false value.
	StyleSheet (see page 165)	Assigns a single stylesheet.
	Text (see page 165)	The HTML text being edited.

	ToolbarLevel (see page 166)	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 57)", "FormDesign", "XSLTDesign", "Full", "Custom"
	ToolbarResetCall (see page 166)	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 enirely 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 167)	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.

Legend

	Property

HtmlEditor.Archived Property

This signifies that the selected content (see page 337) 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
```

Description

Alomst copy from EkWebControl (see page 117) :: Authenticated

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
```

Description

Alomst copy from EkWebControl (see page 117) :: CmsSitePath

HtmlEditor.ContentAction Property

The action to perform on the posted content (see page 337). 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.ContentTitle Property

The title for the selected Content (see page 337). This is set when new content (see page 337) is read. It can not be change 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 (see page 337) the selected ID refers to. If this value is blank, then the content (see page 337) is a custom type, to be defined type, or unsupported type.

These are the possible types:

HTML Form

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 (see page 337).

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 this is 'true' then the user can upload locally selected images and linked files that they have put into their content (see page 337). If enabled, the automatic upload mechanism will allow upload the local files automatically with minimal user intervention. If this value is 'false' then 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, this will include the information about the content (see page 337) in fields posted with a form post. (This fields are defined in the summary of this control.)

Set the value to 'false' if the page will maintain 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 164) 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
```

Description

Alomst copy from EkWebControl (see page 117) :: IsLoggedIn

HtmlEditor.loggedInUserID Property

DESIGN TIME ONLY: Read only property that lets a developer know the user id of the user who is currenlty looged 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 int 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 Integer
```

HtmlEditor.loggedInUserName Property

DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currenlty looged 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. 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 (see page 337). This can be used to simplify the display or to debug the operation of editing content (see page 337) 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 (see page 337) is encoded then it must be decoded on the recieving 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("Whether 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("Whether to encode the HTML content.")>
Public Property SafeEncode() As bool
```

Example 1

validateRequest="false"

Example 2

@ Page Language="VB" validateRequest="false" AutoEventWireup="false" CodeFile="EditAnyHtmlSample.aspx.vb" Inherits="EditAnyHtmlSample"

HtmlEditor.ShowCMSToolbar Property

Controls (see page 5) whether the CMS toolbar is displayed in the editor's toolbar. If this value is true, then commands containing the 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 this is 'true' then the title of the selected content (see page 337) is displayed. The user can then edit the title. The submission of content (see page 337) posts up the title, even if it is not displayed to the user, unless IncludeContentInfo (see page 161) is set to a false value.

C#

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

```
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue(true)]
[Description("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("Show the content title for edit.")>
Public Property ShowTitle() As bool
```

HtmlEditor.StyleSheet Property

Assigns a single stylesheet.

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.")]
public new string Text;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
```

```
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The HTML text.")>
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 57)", "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 enirely 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;
```

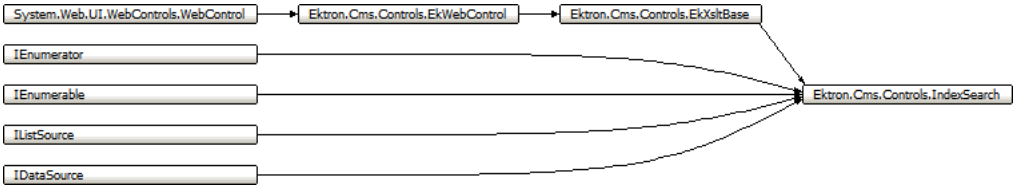
Visual Basic

```
<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
```

IndexSearch Class

Summary description for IndexSearch.

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

Members

Public Methods

	Name	Description
	IndexSearch (see page 171)	Constructor for IndexSearch class.

IndexSearch Structures

	Name	Description
	SearchPram (see page 172)	parameter structure used for search.

Topics

Name	Description
IndexSearch Structures (see page 171)	The structures of the IndexSearch class are listed here.
IndexSearch Methods (see page 172)	The methods of the IndexSearch class are listed here.
IndexSearch Properties (see page 176)	The properties of the IndexSearch class are listed here.

IndexSearch Methods

	Name	Description
	AddParm (see page 173)	
	CopyTo (see page 173)	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 173)	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 174)	Gets a System.Collections.IEnumerator for the collection.
	GetSearchParmDataType (see page 174)	Returns the search parameter data type.
	GetSearchParmSearchType (see page 174)	Returns the search parameter search type.
	GetXPathValues (see page 174)	Returns array of strings using Xpaths.
	MoveNext (see page 175)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 175)	Brings the index of the collection back to zero. Does not clear the collection.
	Search (see page 175)	
	this (see page 176)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	Wrap (see page 176)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some HTML text.

IndexSearch Properties

	Name	Description
	ButtonImgSrc (see page 177)	Optional: The path to the image you wish to use for the button.
	ButtonText (see page 178)	Label text for the text button.
	ContentParameter (see page 178)	Checks the QueryString for this value and replaces the collection with that content (see page 337) block when specified. Leave blank to always display the collection.
	Count (see page 178)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 178)	Gets the current object in the collection.
	EkItems (see page 179)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	EmptyResultMsg (see page 179)	The return message for no hits.
	EncodeContentHtml (see page 179)	Set the property to true if you wish to html encode the Content (see page 337) Html in returned result."
	FolderId (see page 179)	The Path of the folder you wish to start the search. Root starts at "\".
	FormName (see page 180)	The form name.
	HitsMarker (see page 180)	This symbol shows how many search criteria each hit matched
	IsSynchronized (see page 180)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	Labelafter (see page 180)	Specify a label for the date field where search criteria is "After".
	Labelbefore (see page 181)	Specify a label for the date field where search criteria is "Before".
	LabelBetween (see page 181)	Specify label for numeric field where search criteria is based on range of values.
	LabelContains (see page 181)	Specify the label for a string field where search criteria "Contains".
	LabeldateBetween (see page 182)	Specify a label for the date field where search criteria based on a range of dates.
	LabelEqual (see page 182)	Specify label for numeric field where search criteria is "Equal".
	LabelExactPhrase (see page 182)	Specify a label for the string field where search criteria "Exact Phrase".
	LabelGreaterThan (see page 183)	Specify label for numeric field where search criteria is "Greater Than".
	Labelinputdate (see page 183)	Specify a label for the message that appears when date entered are blank/incorrect.
	Labelinputnumber (see page 183)	Specify a label for the message that appears when values entered are blank/incorrect.
	LabelLessThan (see page 184)	Specify label for numeric field where search criteria "Less Than".
	Labelon (see page 184)	Specify a label for the Date field where search criteria is "On".
	LabelSelect (see page 184)	Specify Labels for first item on selection dropdown.
	LinkTarget (see page 185)	Gets or sets the target window for the result to be displayed.
	MaxResults (see page 185)	The total number of results to display. If set to zero, unlimited results will be set.
	OrderBy (see page 185)	Specify the order of the results.
	OrderByDirection (see page 185)	Direction of the result.
	Recursive (see page 186)	Select true if you wish to search the subfolders of the selected FolderPath.
	SearchParmXML (see page 186)	Predefine a set of search criteria.
	ShowSearchBoxAlways (see page 186)	If this property is set to false, the search box will not be visible on PostBack.
	SyncRoot (see page 186)	Gets an object that can be used to synchronize the collection.
	Weighted (see page 187)	Select true if you wish to search the subfolders of the selected FolderPath.
	XmlConfigId (see page 187)	The Path of the folder you wish to start the search. Root starts at "\".
	XmlDoc (see page 187)	Returns the XSLT object for the collection.

Legend

	Constructor
	Structure

	Property

IndexSearch Methods

	Name	Description
	AddParm (see page 173)	
	CopyTo (see page 173)	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 173)	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 174)	Gets a System.Collections.IEnumerator for the collection.
	GetSearchParmDataType (see page 174)	Returns the search parameter data type.
	GetSearchParmSearchType (see page 174)	Returns the search parameter search type.
	GetXPathValues (see page 174)	Returns array of strings using Xpaths.
	MoveNext (see page 175)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 175)	Brings the index of the collection back to zero. Does not clear the collection.
	Search (see page 175)	
	this (see page 176)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	Wrap (see page 176)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some HTML text.

Legend

	Method

IndexSearch Properties

	Name	Description
	ButtonImgSrc (see page 177)	Optional: The path to the image you wish to use for the button.
	ButtonText (see page 178)	Label text for the text button.
	ContentParameter (see page 178)	Checks the QueryString for this value and replaces the collection with that content (see page 337) block when specified. Leave blank to always display the collection.
	Count (see page 178)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 178)	Gets the current object in the collection.
	EkItems (see page 179)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	EmptyResultMsg (see page 179)	The return message for no hits.
	EncodeContentHtml (see page 179)	Set the property to true if you wish to html encode the Content (see page 337) Html in returned result."
	FolderId (see page 179)	The Path of the folder you wish to start the search. Root starts at "\".
	FormName (see page 180)	The form name.
	HitsMarker (see page 180)	This symbol shows how many search criteria each hit matched
	IsSynchronized (see page 180)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	Labelafter (see page 180)	Specify a label for the date field where search criteria is "After".
	Labelbefore (see page 181)	Specify a label for the date field where search criteria is "Before".
	LabelBetween (see page 181)	Specify label for numeric field where search criteria is based on range of values.
	LabelContains (see page 181)	Specify the label for a string field where search criteria "Contains".
	LabeldateBetween (see page 182)	Specify a label for the date field where search criteria based on a range of dates.
	LabelEqual (see page 182)	Specify label for numeric field where search criteria is "Equal".
	LabelExactPhrase (see page 182)	Specify a label for the string field where search criteria "Exact Phrase".

	LabelGreaterThan (see page 183)	Specify label for numeric field where search criteria is "Greater Than".
	Labelinputdate (see page 183)	Specify a label for the message that appears when date entered are blank/incorrect.
	Labelinputnumber (see page 183)	Specify a label for the message that appears when values entered are blank/incorrect.
	LabelLessThan (see page 184)	Specify label for numeric field where search criteria "Less Than".
	LabelOn (see page 184)	Specify a label for the Date field where search criteria is "On".
	LabelSelect (see page 184)	Specify Labels for first item on selection dropdown.
	LinkTarget (see page 185)	Gets or sets the target window for the result to be displayed.
	MaxResults (see page 185)	The total number of results to display. If set to zero, unlimited results will be set.
	OrderBy (see page 185)	Specify the order of the results.
	OrderByDirection (see page 185)	Direction of the result.
	Recursive (see page 186)	Select true if you wish to search the subfolders of the selected FolderPath.
	SearchParmXML (see page 186)	Predefine a set of search criteria.
	ShowSearchBoxAlways (see page 186)	If this property is set to false, the search box will not be visible on PostBack.
	SyncRoot (see page 186)	Gets an object that can be used to synchronize the collection.
	Weighted (see page 187)	Select true if you wish to search the subfolders of the selected FolderPath.
	XmlConfigId (see page 187)	The Path of the folder you wish to start the search. Root starts at "\".
	XmlDoc (see page 187)	Returns the XSLT object for the collection.

Legend

	Property

Public Methods

	Name	Description
	IndexSearch (see page 171)	Constructor for IndexSearch class.

Legend

	Constructor

IndexSearch Structures

	Name	Description
	SearchPram (see page 172)	parameter structure used for search.

Legend

	Structure

IndexSearch.IndexSearch Constructor

Constructor for IndexSearch class.

C#

```
public IndexSearch();
```

Visual Basic

```
Public Sub IndexSearch()
```

IndexSearch Structures

The structures of the IndexSearch class are listed here.

Structures

	Name	Description
	SearchPram (see page 172)	parameter structure used for search.

Legend

	Structure

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 173)	
	CopyTo (see page 173)	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 173)	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 174)	Gets a System.Collections.IEnumerator for the collection.
	GetSearchParmDataType (see page 174)	Returns the search parameter data type.
	GetSearchParmSearchType (see page 174)	Returns the search parameter search type.
	GetXPathValues (see page 174)	Returns array of strings using Xpaths.
	MoveNext (see page 175)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 175)	Brings the index of the collection back to zero. Does not clear the collection.
	Search (see page 175)	
	this (see page 176)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	Wrap (see page 176)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some HTML text.

Legend

	Method

AddParm Method

Overload List

	Name	Description
	IndexSearch.AddParm () (see page 173)	Adds a parameter to the search.
	IndexSearch.AddParm (SearchPram) (see page 173)	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
Parm	param name as SearchPram (see page 172)

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
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
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
XPath	XPath string

Returns

Array of Values

IndexSearch.MoveNext Method

Moves up the index of the Collection (see page 63). 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

Overload List

	Name	Description
	IndexSearch.Search () (see page 175)	Performs the search action.
	IndexSearch.Search (System.Web.HttpRequest) (see page 175)	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
Request	Http Request

IndexSearch.this Indexer

Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) 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
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 177)	Optional: The path to the image you wish to use for the button.
	ButtonText (see page 178)	Label text for the text button.
	ContentParameter (see page 178)	Checks the QueryString for this value and replaces the collection with that content (see page 337) block when specified. Leave blank to always display the collection.
	Count (see page 178)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 178)	Gets the current object in the collection.
	EkItems (see page 179)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	EmptyResultMsg (see page 179)	The return message for no hits.
	EncodeContentHtml (see page 179)	Set the property to true if you wish to html encode the Content (see page 337) Html in returned result."
	FolderId (see page 179)	The Path of the folder you wish to start the search. Root starts at "\".
	FormName (see page 180)	The form name.

	HitsMarker (see page 180)	This symbol shows how many search criteria each hit matched
	IsSynchronized (see page 180)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	Labelafter (see page 180)	Specify a label for the date field where search criteria is "After".
	Labelbefore (see page 181)	Specify a label for the date field where search criteria is "Before".
	LabelBetween (see page 181)	Specify label for numeric field where search criteria is based on range of values.
	LabelContains (see page 181)	Specify the label for a string field where search criteria "Contains".
	LabeldateBetween (see page 182)	Specify a label for the date field where search criteria based on a range of dates.
	LabelEqual (see page 182)	Specify label for numeric field where search criteria is "Equal".
	LabelExactPhrase (see page 182)	Specify a label for the string field where search criteria "Exact Phrase".
	LabelGreaterThan (see page 183)	Specify label for numeric field where search criteria is "Greater Than".
	Labelinputdate (see page 183)	Specify a label for the message that appears when date entered are blank/incorrect.
	Labelinputnumber (see page 183)	Specify a label for the message that appears when values entered are blank/incorrect.
	LabelLessThan (see page 184)	Specify label for numeric field where search criteria "Less Than".
	Labelon (see page 184)	Specify a label for the Date field where search criteria is "On".
	LabelSelect (see page 184)	Specify Labels for first item on selection dropdown.
	LinkTarget (see page 185)	Gets or sets the target window for the result to be displayed.
	MaxResults (see page 185)	The total number of results to display. If set to zero, unlimited results will be set.
	OrderBy (see page 185)	Specify the order of the results.
	OrderByDirection (see page 185)	Direction of the result.
	Recursive (see page 186)	Select true if you wish to search the subfolders of the selected FolderPath.
	SearchParmXML (see page 186)	Predefine a set of search criteria.
	ShowSearchBoxAlways (see page 186)	If this property is set to false, the search box will not be visible on PostBack.
	SyncRoot (see page 186)	Gets an object that can be used to synchronize the collection.
	Weighted (see page 187)	Select true if you wish to search the subfolders of the selected FolderPath.
	XmlConfigId (see page 187)	The Path of the folder you wish to start the search. Root starts at "\".
	XmlDoc (see page 187)	Returns the XSLT object for the collection.

Legend

	Property

IndexSearch.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
```

IndexSearch.ButtonText Property

Label text for the text 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 collection with that content (see page 337) 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.")]
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.")>
Public Property ContentParameter() As string
```

IndexSearch.Count Property

Indicates the number of elements in your Collection (see page 63).

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.EkItems Property

Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). 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 return message for no hits.

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 337) 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 Path of the folder you wish to start the search. Root starts at "\".

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 int 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 Integer
```

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.HitsMarker Property

This symbol shows how many search criteria each hit matched

C#

```
[Category("Ektron"), TypeConverter(typeof(HitsMarkerTypeConverter)), DefaultValue("*")]
[Description("The symbol shows how many search criteria each hit matched")]
public string HitsMarker;
```

Visual Basic

```
<Category("Ektron"), TypeConverter(typeof(HitsMarkerTypeConverter)), DefaultValue("*")>
<Description("The symbol shows how many search criteria each hit matched")>
Public Property HitsMarker() 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 label for 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 label for 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 label for 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 label for 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 label for 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 label for 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 label for date field where search criteria based on range of dates.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_DBTWEEN)]
public string LabeldateBetween;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for date field where search criteria based on 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 label for 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 label for 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 label for 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 label for 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.

C#

```
[Category("Ektron"), DefaultValue("Title"), Description("Specify the order of the results.")]
public string OrderBy;
```

Visual Basic

```
<Category("Ektron"), DefaultValue("Title"), Description("Specify the order of the results.")>
Public Property OrderBy() As string
```

IndexSearch.OrderByDirection Property

Direction of the result.

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 to search the subfolders of the selected FolderPath.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Select true if you wish to do weighted search")]
public bool Weighted;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Select true if you wish to do weighted search")>
Public Property Weighted() As bool
```

IndexSearch.XmlConfigId Property

The Path of the folder you wish to start the search. Root starts at "\".

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Get or Set Xml Configuration ID")]
[Editor(typeof(XMLConfigIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int XmlConfigId;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Get or Set Xml Configuration ID")>
<Editor(typeof(XMLConfigIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property XmlConfigId() As Integer
```

IndexSearch.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

Members

Public Methods

	Name	Description
	LanguageAPI (see page 189)	Upon initialization, SiteLanguage (see page 193) is set to DefaultLanguage (see page 192).

LanguageAPI Methods

	Name	Description
	Fill (see page 190)	Use this function along with LanguageLogic to implement your custom language control logic.
	GetLanguage (see page 190)	Returns a Language string corresponding to a given LanguageID.
	GetLanguageID (see page 190)	Returns the LanguageID corresponding to a given Language string.
	IsValid (see page 190)	

Topics

Name	Description
LanguageAPI Methods (see page 190)	The methods of the LanguageAPI class are listed here.
LanguageAPI Properties (see page 191)	The properties of the LanguageAPI class are listed here.

LanguageAPI Properties

	Name	Description
	CurrentLanguage (see page 191)	Returns the current language (from a cookie).
	CurrentLanguageID (see page 192)	Returns the current language ID (from a cookie).
	DefaultLanguage (see page 192)	If not overridden, the system string "Default" is used.
	DefaultLanguageID (see page 192)	If not overridden, the ID '0' is used.
	LangaugeIdList (see page 192)	Returns the current language ID (from a cookie).
	LangaugeTitleList (see page 192)	Returns the current language ID (from a cookie).
	MultiLanguageEnabled (see page 193)	Returns the current language (from a cookie).
	OverrideDefault (see page 193)	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 193) is not equal to DefaultLanguage (see page 192). If they are equal, false is returned.
	SiteLanguage (see page 193)	Sets the site language of the site. Runs through the IsValid (see page 191) function to verify that the language is an active language in the system.
	SiteLanguageID (see page 193)	If not overridden, the ID '0' is used.

Legend

	Constructor
	virtual
	Property

LanguageAPI Methods

	Name	Description
	Fill (see page 190)	Use this function along with LanguageLogic to implement your custom language control logic.
	GetLanguage (see page 190)	Returns a Language string corresponding to a given LanguageID.
	GetLanguageID (see page 190)	Returns the LanguageID corresponding to a given Language string.
	IsValid (see page 190)	

Legend

	Method
	virtual

LanguageAPI Properties

	Name	Description
	CurrentLanguage (see page 191)	Returns the current language (from a cookie).
	CurrentLanguageID (see page 192)	Returns the current language ID (from a cookie).
	DefaultLanguage (see page 192)	If not overridden, the system string "Default" is used.
	DefaultLanguageID (see page 192)	If not overridden, the ID '0' is used.
	LangaugeIdList (see page 192)	Returns the current language ID (from a cookie).
	LangaugeTitleList (see page 192)	Returns the current language ID (from a cookie).
	MultiLanguageEnabled (see page 193)	Returns the current language (from a cookie).
	OverrideDefault (see page 193)	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 193) is not equal to DefaultLanguage (see page 192). If they are equal, false is returned.
	SiteLanguage (see page 193)	Sets the site language of the site. Runs through the IsValid (see page 191) function to verify that the language is an active language in the system.
	SiteLanguageID (see page 193)	If not overridden, the ID '0' is used.

Legend

	Property

Public Methods

	Name	Description
	LanguageAPI (see page 189)	Upon initialization, SiteLanguage (see page 193) is set to DefaultLanguage (see page 192).

Legend

	Constructor

LanguageAPI.LanguageAPI Constructor

Upon initialization, SiteLanguage (see page 193) is set to DefaultLanguage (see page 192).

C#

```
public LanguageAPI();
```

Visual Basic

```
Public Sub LanguageAPI()
```

LanguageAPI Methods

The methods of the LanguageAPI class are listed here.

Public Methods

	Name	Description
	Fill (see page 190)	Use this function along with LanguageLogic to implement your custom language control logic.
	GetLanguage (see page 190)	Returns a Language string corresponding to a given LanguageID.
	GetLanguageID (see page 190)	Returns the LanguageID corresponding to a given Language string.
	IsValid (see page 190)	

Legend

	Method
	virtual

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

Overload List

	Name	Description
	LanguageAPI.IsValid (int) (see page 191)	Returns true if the passed LanguageID is an active system language. This function is overloaded. See Also: LanguageAPI.IsValid Method (string) (see page 191)
	LanguageAPI.IsValid (string) (see page 191)	Returns true if the passed Language is an active system language. This function is overloaded. See Also: LanguageAPI.IsValid Method (int) (see page 191)

LanguageAPI.IsValid Method (int)

Returns true if the passed LanguageID is an active system language.

This function is overloaded. See Also: LanguageAPI.IsValid Method (string) (see page 191)

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.

This function is overloaded. See Also: LanguageAPI.IsValid Method (int) (see page 191)

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 191)	Returns the current language (from a cookie).
	CurrentLanguageID (see page 192)	Returns the current language ID (from a cookie).
	DefaultLanguage (see page 192)	If not overridden, the system string "Default" is used.
	DefaultLanguageID (see page 192)	If not overridden, the ID '0' is used.
	LangaugeIdList (see page 192)	Returns the current language ID (from a cookie).
	LangaugeTitleList (see page 192)	Returns the current language ID (from a cookie).
	MultiLanguageEnabled (see page 193)	Returns the current language (from a cookie).
	OverrideDefault (see page 193)	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 193) is not equal to DefaultLanguage (see page 192). If they are equal, false is returned.
	SiteLanguage (see page 193)	Sets the site language of the site. Runs through the IsValid (see page 191) function to verify that the language is an active language in the system.
	SiteLanguageID (see page 193)	If not overridden, the ID '0' is used.

Legend

	Property

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 the current language ID (from a cookie).

C#

```
[Browsable(false)]
public int LangaugeIdList;
```

Visual Basic

```
<Browsable(false)>
Public Property LangaugeIdList() As Integer
```

LanguageAPI.LangaugeTitleList Property

Returns the current language ID (from a cookie).

C#

```
[Browsable(false)]
public string LangaugeTitleList;
```

Visual Basic

```
<Browsable(false)>
Public Property LangaugeTitleList() As string
```

LanguageAPI.MultiLanguageEnabled Property

Returns the current language (from a cookie).

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 193) is not equal to DefaultLanguage (see page 192). 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 191) 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

Summary description for LanguageSelect server control.

Class Hierarchy

C#

```
public class LanguageSelect : EkWebControl;
```

Visual Basic

```
Public Class LanguageSelect
Inherits EkWebControl
```

File

LanguageSelect.cs

Members

LanguageSelect Methods

	Name	Description
	Fill (see page 195)	Method that can be overridden and used to grab data from the back end for each server control.
	Wrap (see page 195)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
LanguageSelect Methods (see page 195)	The methods of the LanguageSelect class are listed here.
LanguageSelect Properties (see page 195)	The properties of the LanguageSelect class are listed here.

LanguageSelect Properties

	Name	Description
	LabelName (see page 196)	Label next to the language select.
	Text (see page 196)	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.

Legend

	Method
	Property

LanguageSelect Methods

	Name	Description
	Fill (see page 195)	Method that can be overridden and used to grab data from the back end for each server control.
	Wrap (see page 195)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

LanguageSelect Properties

	Name	Description
	LabelName (see page 196)	Label next to the language select.
	Text (see page 196)	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.

Legend

	Property

LanguageSelect Methods

The methods of the LanguageSelect class are listed here.

Public Methods

	Name	Description
	Fill (see page 195)	Method that can be overridden and used to grab data from the back end for each server control.
	Wrap (see page 195)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
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 196)	Label next to the language select.

	Text (see page 196)	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.

Legend

	Property

LanguageSelect.LabelName Property

Label next to the language select.

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;
```

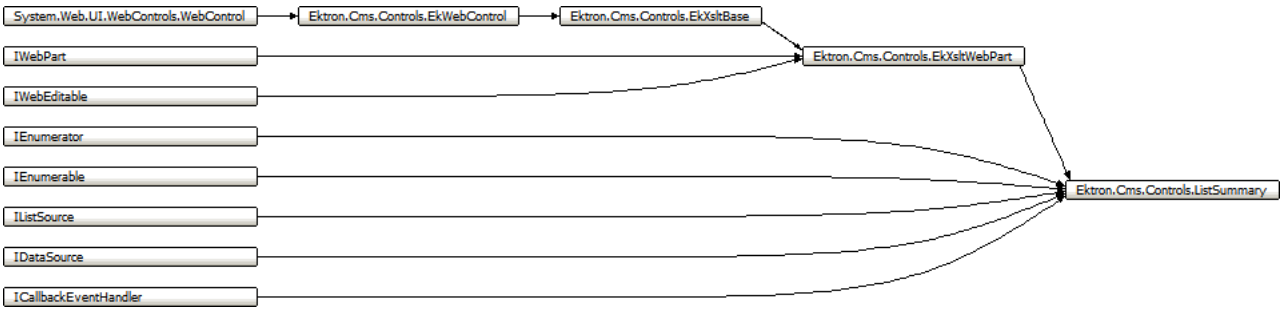
Visual Basic

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

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

Members

ListSummary Methods

	Name	Description
	CopyTo (see page 199)	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 199)	Populates the data for the ListSummary.
	GetEnumerator (see page 199)	Gets a System.Collections.IEnumerator for the ListSummary.
	GetRssFeed (see page 200)	This method returns the ListSummary RSS feed.
	GetView (see page 200)	Gets the view data using dataset.
	GetViewNames (see page 200)	Gets the view names.
	MoveNext (see page 200)	Move to next index.
	Reset (see page 201)	Resets the index.
	this (see page 201)	Returns the current ContentItem.ContentRow.
	ToString (see page 201)	Overrides the system ToString function.
	Wrap (see page 201)	Wrap the Html toolbar.

Topics

Name	Description
ListSummary Methods (see page 199)	The methods of the ListSummary class are listed here.
ListSummary Properties (see page 202)	The properties of the ListSummary class are listed here.

ListSummary Properties

	Name	Description
	CacheInterval (see page 202)	Sets the cache interval in second(s).
	ContentParameter (see page 203)	Checks the QueryString for this value and replaces the collection with that content (see page 337) block when specified. Leave blank to always display the collection.
	ContentType (see page 203)	The XSLT to use to render the Collection (see page 63) (optional).
	Count (see page 203)	Indicates the number of elements in your ListSummary.
	Current (see page 203)	Gets the current index.
	Direction (see page 204)	Direction for sorting results, "ascending" or "descending".
	EkItems (see page 204)	Returns the array of ContentBase.
	EnablePaging (see page 204)	Enables paging if the total records exceeds MaxResult value.
	FolderID (see page 205)	to be changed
	FolderPath (see page 205)	to be changed
	GetHtml (see page 205)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	IncludeIcons (see page 206)	Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.
	IsSynchronized (see page 206)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 206)	Gets or Sets the target window for the result.

	MaxResults (see page 207)	Maximum number of results to be returned.
	OrderKey (see page 207)	Sort order key.
	Random (see page 207)	If set to true, grabs a single content (see page 337) block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the collection.
	Recursive (see page 208)	Determines if subfolders of the selected folders should be brought into the list.
	SyncRoot (see page 208)	Gets an object that can be used to synchronize the ListSummary.
	XmlDoc (see page 208)	Returns the Xml object loaded for ListSummary.

Legend

	Method
	Property
	virtual

ListSummary Methods

	Name	Description
	CopyTo (see page 199)	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 199)	Populates the data for the ListSummary.
	GetEnumerator (see page 199)	Gets a System.Collections.IEnumerator for the ListSummary.
	GetRssFeed (see page 200)	This method returns the ListSummary RSS feed.
	GetView (see page 200)	Gets the view data using dataset.
	GetViewNames (see page 200)	Gets the view names.
	MoveNext (see page 200)	Move to next index.
	Reset (see page 201)	Resets the index.
	this (see page 201)	Returns the current ContentItem.ContentRow.
	ToString (see page 201)	Overrides the system ToString function.
	Wrap (see page 201)	Wrap the Html toolbar.

Legend

	Method

ListSummary Properties

	Name	Description
	CacheInterval (see page 202)	Sets the cache interval in second(s).
	ContentParameter (see page 203)	Checks the QueryString for this value and replaces the collection with that content (see page 337) block when specified. Leave blank to always display the collection.
	ContentType (see page 203)	The XSLT to use to render the Collection (see page 63) (optional).
	Count (see page 203)	Indicates the number of elements in your ListSummary.
	Current (see page 203)	Gets the current index.
	Direction (see page 204)	Direction for sorting results, "ascending" or "descending".
	EkItems (see page 204)	Returns the array of ContentBase.
	EnablePaging (see page 204)	Enables paging if the total records exceeds MaxResult value.
	FolderID (see page 205)	to be changed
	FolderPath (see page 205)	to be changed
	GetHtml (see page 205)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	IncludeIcons (see page 206)	Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.
	IsSynchronized (see page 206)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 206)	Gets or Sets the target window for the result.
	MaxResults (see page 207)	Maximum number of results to be returned.
	OrderKey (see page 207)	Sort order key.

	Random (see page 207)	If set to true, grabs a single content (see page 337) block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the collection.
	Recursive (see page 208)	Determines if subfolders of the selected folders should be brought into the list.
	SyncRoot (see page 208)	Gets an object that can be used to synchronize the ListSummary.
	XmlDoc (see page 208)	Returns the Xml object loaded for ListSummary.

Legend

	Property
	virtual

ListSummary Methods

The methods of the ListSummary class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 199)	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 199)	Populates the data for the ListSummary (see page 196).
	GetEnumerator (see page 199)	Gets a System.Collections.IEnumerator for the ListSummary (see page 196).
	GetRssFeed (see page 200)	This method returns the ListSummary (see page 196) RSS feed.
	GetView (see page 200)	Gets the view data using dataset.
	GetViewNames (see page 200)	Gets the view names.
	MoveNext (see page 200)	Move to next index.
	Reset (see page 201)	Resets the index.
	this (see page 201)	Returns the current ContentItem.ContentRow.
	ToString (see page 201)	Overrides the system ToString function.
	Wrap (see page 201)	Wrap the Html toolbar.

Legend

	Method

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 196).

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 196).

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 196) 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
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
i	Index value

Returns

Content (see page 337) 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
Html	

ListSummary Properties

The properties of the ListSummary class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 202)	Sets the cache interval in second(s).
	ContentParameter (see page 203)	Checks the QueryString for this value and replaces the collection with that content (see page 337) block when specified. Leave blank to always display the collection.
	ContentType (see page 203)	The XSLT to use to render the Collection (see page 63) (optional).
	Count (see page 203)	Indicates the number of elements in your ListSummary (see page 196).
	Current (see page 203)	Gets the current index.
	Direction (see page 204)	Direction for sorting results, "ascending" or "descending".
	EkItems (see page 204)	Returns the array of ContentBase.
	EnablePaging (see page 204)	Enables paging if the total records exceeds MaxResult value.
	FolderID (see page 205)	to be changed
	FolderPath (see page 205)	to be changed
	GetHtml (see page 205)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	IncludeIcons (see page 206)	Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.
	IsSynchronized (see page 206)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 206)	Gets or Sets the target window for the result.
	MaxResults (see page 207)	Maximum number of results to be returned.
	OrderKey (see page 207)	Sort order key.
	Random (see page 207)	If set to true, grabs a single content (see page 337) block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the collection.
	Recursive (see page 208)	Determines if subfolders of the selected folders should be brought into the list.
	SyncRoot (see page 208)	Gets an object that can be used to synchronize the ListSummary (see page 196).
	XmlDoc (see page 208)	Returns the Xml object loaded for ListSummary (see page 196).

Legend

	Property
	virtual

ListSummary.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()]
public virtual double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("interval to cache this control in second(s)")>
```

```
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

ListSummary.ContentParameter Property

Checks the QueryString for this value and replaces the collection with that content (see page 337) 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()]
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()>
Public Property ContentParameter() As string
```

ListSummary.ContentType Property

The XSLT to use to render the Collection (see page 63) (optional).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Detemines the type for content to returns. example is forms")]
[Personalizable()]
public Ektron.Cms.Controls.CmsWebService.CMSContentType ContentType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Detemines the type for content to returns. example is forms")>
<Personalizable()>
Public Property ContentType() As Ektron.Cms.Controls.CmsWebService.CMSContentType
```

ListSummary.Count Property

Indicates the number of elements in your ListSummary (see page 196).

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#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("\\")]
[Description("The direction (ascending or descending) that you wish your sorted results to be
displayed in.")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Direction"),
WebDescription("Direction")]
public Ektron.Cms.Controls.CmsWebService.OrderByDirection Direction;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("\\")>
<Description("The direction (ascending or descending) that you wish your sorted results to be
displayed in.")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Direction"),
WebDescription("Direction")>
Public 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()]
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()>
Public Property EnablePaging() As bool
```

ListSummary.FolderID Property

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()]
public int 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()>
Public Property FolderID() As Integer
```

Description

to be changed

ListSummary.FolderPath Property

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Browsable(false)]
[Description("The Path of the folder you wish to select data from. 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 you wish to select data from. Root starts at \"\\\".")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Obsolete()>
Public Property FolderPath() As string
```

Description

to be changed

ListSummary.GetHtml Property

If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) 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()]
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()>
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()]
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()>
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.MaxResults Property

Maximum number of results to be returned.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[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(0)>
<Description("The maximum number of results to fetch. If set to zero, there is no limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

ListSummary.OrderKey Property

Sort order key.

C#

```
[Bindable(true)]
[DefaultValue(Ektron.Cms.Controls.CmsWebService.TeasersOrderBy.Title)]
[Description("The key you wish to sort your results by.")]
[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 you wish to sort your results by.")>
<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 (see page 337) block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the collection.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Set this property to True to grab a single, random piece of content from this
collection.")]
[Personalizable()]
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
collection.")>
<Personalizable()>
Public Property Random() As bool
```

ListSummary.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.")]
[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 grab results in subfolders of your
FolderPath.")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Recursive"), WebDescription("Recursive:
")>
Public Property Recursive() As bool
```

ListSummary.SyncRoot Property

Gets an object that can be used to synchronize the ListSummary (see page 196).

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 196).

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

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

Login Class

Summary description for Login.

Class Hierarchy

C#

```
public class Login : EkWebControl;
```

Visual Basic

```
Public Class Login
Inherits EkWebControl
```

File

Login.cs

Members

Login Methods

	Name	Description
	Fill (see page 210)	Method that can be overridden and used to grab data from the back end for each server control.

Topics

Name	Description
Login Methods (see page 210)	The methods of the Login class are listed here.
Login Properties (see page 210)	The properties of the Login class are listed here.

Login Properties

	Name	Description
	AutoAddType (see page 210)	Set this property to define auto-added Active Directory users as members or authors.
	AutoLogin (see page 211)	Set this property to true if you want to use Auto Login with Active Directory.
	OnlyAllowMemberLogin (see page 211)	Set this property to true if you only allow membership users.
	Password (see page 211)	Write Only value for passing in the password.
	PromptLogout (see page 211)	Set this property to true if you only allow membership users.
	SuppressHelpButton (see page 211)	Set this property to true if you don't want to show the Help button.
	Text (see page 212)	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 212)	Write Only value for passing in the UserName.

Legend

	Method
	Property

Login Methods

	Name	Description
	Fill (see page 210)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

Login Properties

	Name	Description
	AutoAddType (see page 210)	Set this property to define auto-added Active Directory users as members or authors.
	AutoLogin (see page 211)	Set this property to true if you want to use Auto Login with Active Directory.
	OnlyAllowMemberLogin (see page 211)	Set this property to true if you only allow membership users.
	Password (see page 211)	Write Only value for passing in the password.
	PromptLogout (see page 211)	Set this property to true if you only allow membership users.
	SuppressHelpButton (see page 211)	Set this property to true if you don't want to show the Help button.

	Text (see page 212)	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 212)	Write Only value for passing in the UserName.

Legend

	Property

Login Methods

The methods of the Login class are listed here.

Public Methods

	Name	Description
	Fill (see page 210)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

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 210)	Set this property to define auto-added Active Directory users as members or authors.
	AutoLogin (see page 211)	Set this property to true if you want to use Auto Login with Active Directory.
	OnlyAllowMemberLogin (see page 211)	Set this property to true if you only allow membership users.
	Password (see page 211)	Write Only value for passing in the password.
	PromptLogout (see page 211)	Set this property to true if you only allow membership users.
	SuppressHelpButton (see page 211)	Set this property to true if you don't want to show the Help button.
	Text (see page 212)	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 212)	Write Only value for passing in the UserName.

Legend

	Property

Login.AutoAddType Property

Set this property to define auto-added Active Directory users as members or authors.

C#

```
[Bindable(false), Category("Ektron"), Description("Set this property to define auto-added
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
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 with Active Directory.

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 true if you only allow membership users.

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
```

Membership Class

Using this control users can register to the site.

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

Members

Membership Fields

	Name	Description
	IsPostBack (see page 214)	Sets IsPostBack to false.

Topics

Name	Description
Membership Fields (see page 214)	The fields of the Membership class are listed here.
Membership Methods (see page 214)	The methods of the Membership class are listed here.
Membership Properties (see page 215)	The properties of the Membership class are listed here.

Membership Methods

	Name	Description
	Fill (see page 215)	Method that can be overridden and used to grab data from the back end for each server control.

Membership Properties

	Name	Description
	BoardID (see page 215)	Specify the board ID to use for the terms and conditions text.
	DisplayMode (see page 216)	Choose the control mode. Each mode has it own UI and behaves differently.
	RedirectFailedURL (see page 216)	Redirect the user if registering to the site fails.
	RedirectSuccessURL (see page 216)	Redirect the user after successfully registering to the site.
	RegisterButtonImg (see page 216)	Specify the image to use instead of the register button. The image path should be relative to the root.
	RegisterButtonText (see page 217)	Specify the form submit button text. The default value is "Register".
	ResetButtonImg (see page 217)	Specify the image to use instead of the reset button. The image path should be relative to the root.
	ResetButtonText (see page 217)	Specify the form reset button text. The default is "Reset".
	ShowExtended (see page 217)	Include user's custom properties defined in the CMS. The default value is "True".
	ShowTerms (see page 218)	Display the Terms and Conditions text. The default value is "False".
	Text (see page 218)	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 218)	User already exists error message. The default value is "Username(email) already exists!".
	UserID (see page 218)	Specify the user ID to retrieve information.
	UserSuccessMessage (see page 218)	User successfully registered message. The default value is "You have registered successfully".
	UserUpdateSuccessMessage (see page 219)	Message explaining the user successfully update his/her information. The default value is "You have successfully updated your information".

Legend

	Data Member
	Method
	Property

Membership Fields

	Name	Description
	IsPostBack (see page 214)	Sets IsPostBack to false.

Legend

	Data Member

Membership Methods

	Name	Description
	Fill (see page 215)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

Membership Properties

	Name	Description
	BoardID (see page 215)	Specify the board ID to use for the terms and conditions text.
	DisplayMode (see page 216)	Choose the control mode. Each mode has it own UI and behaves differently.
	RedirectFailedURL (see page 216)	Redirect the user if registering to the site fails.
	RedirectSuccessURL (see page 216)	Redirect the user after successfully registering to the site.
	RegisterButtonImg (see page 216)	Specify the image to use instead of the register button. The image path should be relative to the root.
	RegisterButtonText (see page 217)	Specify the form submit button text. The default value is "Register".
	ResetButtonImg (see page 217)	Specify the image to use instead of the reset button. The image path should be relative to the root.
	ResetButtonText (see page 217)	Specify the form reset button text. The default is "Reset".
	ShowExtended (see page 217)	Include user's custom properties defined in the CMS. The default value is "True".
	ShowTerms (see page 218)	Display the Terms and Conditions text. The default value is "False".
	Text (see page 218)	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 218)	User already exists error message. The default value is "Username(email) already exists!".
	UserID (see page 218)	Specify the user ID to retrieve information.
	UserSuccessMessage (see page 218)	User successfully registered message. The default value is "You have registered successfully".
	UserUpdateSuccessMessage (see page 219)	Message explaining the user successfully update his/her information. The default value is "You have successfully updated your information".

Legend

	Property

Membership Fields

The fields of the Membership class are listed here.

Public Fields

	Name	Description
	IsPostBack (see page 214)	Sets IsPostBack to false.

Legend

	Data Member

Membership.IsPostBack Field

Sets IsPostBack to false.

C#

```
public bool IsPostBack = false;
```

Visual Basic

```
Public IsPostBack As bool = false
```

Membership Methods

The methods of the Membership class are listed here.

Public Methods

	Name	Description
	Fill (see page 215)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

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
	BoardID (see page 215)	Specify the board ID to use for the terms and conditions text.
	DisplayMode (see page 216)	Choose the control mode. Each mode has it own UI and behaves differently.
	RedirectFailedURL (see page 216)	Redirect the user if registering to the site fails.
	RedirectSuccessURL (see page 216)	Redirect the user after successfully registering to the site.
	RegisterButtonImg (see page 216)	Specify the image to use instead of the register button. The image path should be relative to the root.
	RegisterButtonText (see page 217)	Specify the form submit button text. The default value is "Register".
	ResetButtonImg (see page 217)	Specify the image to use instead of the reset button. The image path should be relative to the root.
	ResetButtonText (see page 217)	Specify the form reset button text. The default is "Reset".
	ShowExtended (see page 217)	Include user's custom properties defined in the CMS. The default value is "True".
	ShowTerms (see page 218)	Display the Terms and Conditions text. The default value is "False".
	Text (see page 218)	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 218)	User already exists error message. The default value is "Username(email) already exists!".
	UserID (see page 218)	Specify the user ID to retrieve information.
	UserSuccessMessage (see page 218)	User successfully registered message. The default value is "You have registered successfully".
	UserUpdateSuccessMessage (see page 219)	Message explaining the user successfully update his/her information. The default value is "You have successfully updated your information".

Legend

	Property

Membership.BoardID Property

Specify the board ID to use for the terms and conditions text.

C#

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

```
[Category("Ektron")]
[DefaultValue("\\")]
[Description("The ID of the forum you wish show terms and conditions for.")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int BoardID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("\\")>
<Description("The ID of the forum you wish show terms and conditions for.")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BoardID() As Integer
```

Membership.DisplayMode Property

Choose the control mode. Each mode has it own UI and behaves differently.

C#

```
[Bindable(true), Category("Ektron"), Description("Choose the control mode. Each mode has it
own UI and behaves differently."), DefaultValue(0)]
public Mode DisplayMode;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Choose the control mode. Each mode has it
own UI and behaves differently."), DefaultValue(0)>
Public Property DisplayMode() As Mode
```

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. The default value is "Register".

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. The default is "Reset".

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.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."), DefaultValue(true)]
public bool ShowExtended;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Include user's custom properties defined in
the CMS."), 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(""), DefaultValue(false)]
public bool ShowTerms;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description(""), DefaultValue(false)>
Public Property ShowTerms() As bool
```

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 already exists error message. The default value is "Username(email) already exists!".

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 to retrieve information.

C#

```
[Bindable(false), Browsable(false), Description("Specify the user ID to retrieve
information."), DefaultValue("0")]
public int UserID;
```

Visual Basic

```
<Bindable(false), Browsable(false), Description("Specify the user ID to retrieve
information."), DefaultValue("0")>
Public Property UserID() As Integer
```

Membership.UserSuccessMessage Property

User successfully registered message. The default value is "You have registered successfully".

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. The default value is "You have successfully updated your 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

Ektron Menu class.

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

Members

Menu Methods

	Name	Description
	Fill (see page 221)	Calls up the database and updates the contents of the Menu if the data is out of sync with the specified parameters.
	Wrap (see page 221)	Method that can overridden and be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
Menu Methods (see page 220)	The methods of the Menu class are listed here.
Menu Properties (see page 221)	The properties of the Menu class are listed here.

Menu Properties

	Name	Description
	CacheInterval (see page 221)	Sets the cache interval in second(s).
	DefaultMenuID (see page 222)	Gets or Sets the Default MenuID for this Control.
	DisplayXslt (see page 222)	The Xslt to use to render the Menu (optional)
	DynamicParameter (see page 222)	The parameter to query for a Dynamic Menu.
	Stylesheet (see page 223)	Specify the style sheet relative to site root. Leave blank for default.

Legend

	Method
	Property
	virtual

Menu Methods

	Name	Description
	Fill (see page 221)	Calls up the database and updates the contents of the Menu if the data is out of sync with the specified parameters.
	Wrap (see page 221)	Method that can overridden and be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

Menu Properties

	Name	Description
	CacheInterval (see page 221)	Sets the cache interval in second(s).
	DefaultMenuID (see page 222)	Gets or Sets the Default MenuID for this Control.
	DisplayXslt (see page 222)	The Xslt to use to render the Menu (optional)
	DynamicParameter (see page 222)	The parameter to query for a Dynamic Menu.
	Stylesheet (see page 223)	Specify the style sheet relative to site root. Leave blank for default.

Legend

	Property
	virtual

Menu Methods

The methods of the Menu class are listed here.

Public Methods

	Name	Description
	Fill (see page 221)	Calls up the database and updates the contents of the Menu if the data is out of sync with the specified parameters.
	Wrap (see page 221)	Method that can overridden and be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

Menu.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
```

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
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 221)	Sets the cache interval in second(s).
	DefaultMenuID (see page 222)	Gets or Sets the Default MenuID for this Control.
	DisplayXslt (see page 222)	The Xslt to use to render the Menu (optional)
	DynamicParameter (see page 222)	The parameter to query for a Dynamic Menu.
	Stylesheet (see page 223)	Specify the style sheet relative to site root. Leave blank for default.

Legend

	Property
	virtual

Menu.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)]
```

```
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
```

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 int 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 Integer
```

Menu.DisplayXslt Property

The Xslt to use to render the Menu (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.

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
```

MetaData Class

Summary description for MetaData.

Class Hierarchy

C#

```
public class MetaData : Ektron.Cms.Controls.EkWebControl;
```

Visual Basic

```
Public Class MetaData
Inherits Ektron.Cms.Controls.EkWebControl
```

File

MetaData.cs

Members

MetaData Methods

	Name	Description
	Fill (see page 224)	Method that can be overridden and used to grab data from the back end for each server control.

Topics

Name	Description
MetaData Methods (see page 224)	The methods of the MetaData class are listed here.
MetaData Properties (see page 224)	The properties of the MetaData class are listed here.

MetaData Properties

	Name	Description
	DefaultContentID (see page 224)	to be changed

Legend

	Method
	Property

MetaData Methods

	Name	Description
	Fill (see page 224)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

MetaData Properties

	Name	Description
	DefaultContentID (see page 224)	to be changed

Legend

	Property

MetaData Methods

The methods of the MetaData class are listed here.

Public Methods

	Name	Description
	Fill (see page 224)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

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
	DefaultContentID (see page 224)	to be changed

Legend

	Property

MetaData.DefaultContentID Property

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 int 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 Integer
```

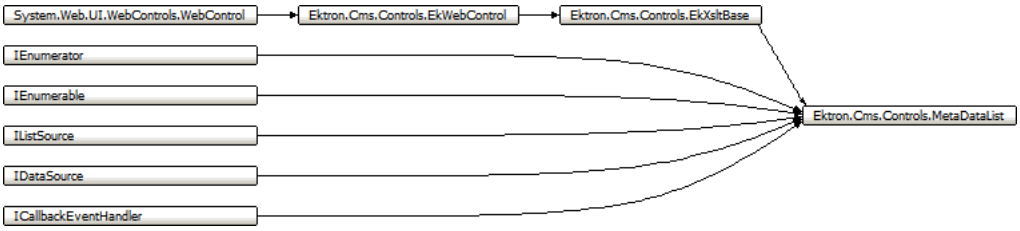
Description

to be changed

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

Members

MetaDataList Methods

	Name	Description
	CopyTo (see page 228)	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 228)	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 228)	Gets a System.Collections.IEnumerator for the collection.
	MoveNext (see page 228)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 228)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 229)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	Wrap (see page 229)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
MetaDataList Methods (see page 227)	The methods of the MetaDataList class are listed here.
MetaDataList Properties (see page 229)	The properties of the MetaDataList class are listed here.

MetaDataList Properties

	Name	Description
	CacheInterval (see page 230)	Sets the cache interval in second(s).
	ContentType (see page 230)	Gets or sets the ContentType for this control.
	Count (see page 230)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 230)	Gets the current object in the collection.
	EkItems (see page 231)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	EnablePaging (see page 231)	Enables paging if the total records exceeds the MaxResult value.
	ExactPhrase (see page 231)	Gets or Sets the boolean value of ExactPhrase.
	FolderID (see page 232)	Gets or sets the dynamic folder ID.
	GetHtml (see page 232)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	IncludeIcons (see page 232)	Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.
	IsSynchronized (see page 233)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	KeyWordName (see page 233)	Gets or Sets KeyWordName for this control.
	KeyWordValue (see page 233)	Gets or sets the KeyWordValue.
	LinkTarget (see page 233)	Gets or Sets the target window for the result.
	MaxNumber (see page 234)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	OrderBy (see page 234)	Order to return the metalist.
	Recursive (see page 234)	If set to true, It will carry out a recursive find from the folderID and all it's children folders. Set to false for only content (see page 337) in that folder.
	SortOrder (see page 235)	Gets or sets the sort order.
	SyncRoot (see page 235)	Gets an object that can be used to synchronize the collection.
	XmlDoc (see page 235)	Returns the XSLT object for metalist control.

Legend

	Method
	Property
	virtual

MetaDataList Methods

	Name	Description
	CopyTo (see page 228)	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 228)	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 228)	Gets a System.Collections.IEnumerator for the collection.
	MoveNext (see page 228)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 228)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 229)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	Wrap (see page 229)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

MetaDataList Properties

	Name	Description
	CacheInterval (see page 230)	Sets the cache interval in second(s).
	ContentType (see page 230)	Gets or sets the ContentType for this control.
	Count (see page 230)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 230)	Gets the current object in the collection.
	EkItems (see page 231)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	EnablePaging (see page 231)	Enables paging if the total records exceeds the MaxResult value.
	ExactPhrase (see page 231)	Gets or Sets the boolean value of ExactPhrase.
	FolderID (see page 232)	Gets or sets the dynamic folder ID.
	GetHtml (see page 232)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	IncludeIcons (see page 232)	Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.
	IsSynchronized (see page 233)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	KeyWordName (see page 233)	Gets or Sets KeyWordName for this control.
	KeyWordValue (see page 233)	Gets or sets the KeyWordValue.
	LinkTarget (see page 233)	Gets or Sets the target window for the result.
	MaxNumber (see page 234)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	OrderBy (see page 234)	Order to return the metalist.
	Recursive (see page 234)	If set to true, It will carry out a recursive find from the folderID and all it's children folders. Set to false for only content (see page 337) in that folder.
	SortOrder (see page 235)	Gets or sets the sort order.
	SyncRoot (see page 235)	Gets an object that can be used to synchronize the collection.
	XmlDoc (see page 235)	Returns the XSLT object for metalist control.

Legend

	Property
	virtual

MetaDataList Methods

The methods of the MetaDataList class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 228)	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 228)	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 228)	Gets a System.Collections.IEnumerator for the collection.
	MoveNext (see page 228)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 228)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 229)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	Wrap (see page 229)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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 63). 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 63) 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
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 230)	Sets the cache interval in second(s).
	ContentType (see page 230)	Gets or sets the ContentType for this control.
	Count (see page 230)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 230)	Gets the current object in the collection.
	EkItems (see page 231)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	EnablePaging (see page 231)	Enables paging if the total records exceeds the MaxResult value.
	ExactPhrase (see page 231)	Gets or Sets the boolean value of ExactPhrase.
	FolderID (see page 232)	Gets or sets the dynamic folder ID.
	GetHtml (see page 232)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	IncludeIcons (see page 232)	Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.
	IsSynchronized (see page 233)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	KeyWordName (see page 233)	Gets or Sets KeyWordName for this control.
	KeyWordValue (see page 233)	Gets or sets the KeyWordValue.
	LinkTarget (see page 233)	Gets or Sets the target window for the result.
	MaxNumber (see page 234)	If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.
	OrderBy (see page 234)	Order to return the metalist.

	Recursive (see page 234)	If set to true, It will carry out a recursive find from the folderID and all it's children folders. Set to false for only content (see page 337) in that folder.
	SortOrder (see page 235)	Gets or sets the sort order.
	SyncRoot (see page 235)	Gets an object that can be used to synchronize the collection.
	XmlDoc (see page 235)	Returns the XSLT object for metalist control.

Legend

	Property
	virtual

MetaDataList.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)]
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
```

MetaDataList.ContentType Property

Gets or sets the ContentType for this control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Detemines the type for content to returns. example is forms")]
public Ektron.Cms.Controls.CmsWebService.CMSContentType ContentType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Detemines the type for content to returns. example is forms")>
Public Property ContentType() As Ektron.Cms.Controls.CmsWebService.CMSContentType
```

MetaDataList.Count Property

Indicates the number of elements in your Collection (see page 63).

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 63). 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

Enables paging if the total records exceeds the 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 MaxNumbers.")]
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 MaxNumbers.")>
Public Property EnablePaging() As bool
```

MetaDataList.ExactPhrase Property

Gets or Sets the boolean value of ExactPhrase.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("True")]
[Description("The value that is determines if the KeyWordValue is in the Metadata list(false)
or if it is the only value in the list(true)")]
public bool ExactPhrase;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("True")>
<Description("The value that is determines if the KeyWordValue is in the Metadata list(false)
or if it is the only value in the list(true)")>
Public Property ExactPhrase() As bool
```

MetaDataList.FolderID Property

Gets or sets the dynamic folder ID.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[Description("This is the root folder to start from")]
[Editor(typeof(FolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("None - Use Default")>
<Description("This is the root folder to start from")>
<Editor(typeof(FolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderID() As Integer
```

MetaDataList.GetHtml Property

If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) 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.")]
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.")>
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.

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

Gets or Sets KeyWordName for this control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(" -None -")]
[Description("Gets or Sets the Default KeyWordName for this Control.")]
[Editor(typeof(MetaDataUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public string KeyWordName;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(" -None -")>
<Description("Gets or Sets the Default KeyWordName for this Control.")>
<Editor(typeof(MetaDataUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property KeyWordName() As string
```

MetaDataList.KeyWordValue Property

Gets or sets the KeyWordValue.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[Description("The value that is assicated with the KeyWord")]
public string KeyWordValue;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("None - Use Default")>
<Description("The value that is assicated with the KeyWord")>
Public Property KeyWordValue() As string
```

MetaDataList.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
```

MetaDataList.MaxNumber Property

If set to true, retrieves the content (see page 337) (Html body) of every content (see page 337) block in the collection for display.

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

Order to return the metalist.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("This value determines how to order the return list")]
public Ektron.Cms.Controls.CmsWebService.ContentOrderBy OrderBy;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("This value determines how to order the return list")>
Public Property OrderBy() As Ektron.Cms.Controls.CmsWebService.ContentOrderBy
```

MetaDataList.Recursive Property

If set to true, It will carry out a recursive find from the folderID and all it's children folders. Set to false for only content (see page 337) in that folder.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, grabs a all content in the folder and all childfolders with that
meta relateionship. Set to false for that folder only.")]
public bool Recursive;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true, grabs a all content in the folder and all childfolders with that
meta relateionship. Set to false for that folder only.")>
Public Property Recursive() As bool
```

MetaDataList.SortOrder Property

Gets or sets the sort order.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("This value determines how to order the return list")]
public Ektron.Cms.Controls.CmsWebService.OrderByDirection SortOrder;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("This value determines how to order the return list")>
Public Property SortOrder() As Ektron.Cms.Controls.CmsWebService.OrderByDirection
```

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 metalist control.

C#

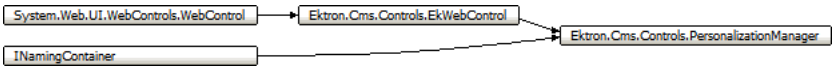
```
public override XmlDocument XmlDoc;
```

Visual Basic

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

PersonalizationManager Class

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

Description

This is class Ektron.Cms.Controls.PersonalizationManager.

Members

PersonalizationManager Methods

	Name	Description
	Fill (see page 237)	Method that can be overridden and used to grab data from the back end for each server control.

Topics

Name	Description
PersonalizationManager Methods (see page 237)	The methods of the PersonalizationManager class are listed here.
PersonalizationManager Properties (see page 237)	The properties of the PersonalizationManager class are listed here.

PersonalizationManager Properties

	Name	Description
	AddContentExitText (see page 238)	The text of the reset link.
	AddContentText (see page 238)	The text of the reset link.
	CurrentUserButtonText (see page 238)	The text of the reset link.
	Display (see page 239)	The display format of the image - horizontal or vertical.
	HeaderBackColor (see page 239)	`HeaderBackColor` sets the header background color.
	HeaderText (see page 239)	The HeaderText of the control.
	PersonalizeExitText (see page 240)	The text of the reset link.
	PersonalizeScopeText (see page 240)	The text of the reset link.
	PersonalizeStartText (see page 240)	The text of the reset link.
	PublicUserButtonText (see page 241)	The text of the reset link.
	ResetLinkText (see page 241)	The text of the reset link.
	ResetLinkToolTip (see page 242)	The text of the reset link tooltip.

Legend

	Method
	Property

PersonalizationManager Methods

	Name	Description
	Fill (see page 237)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

PersonalizationManager Properties

	Name	Description
	AddContentExitText (see page 238)	The text of the reset link.
	AddContentText (see page 238)	The text of the reset link.

	CurrentUserButtonText (see page 238)	The text of the reset link.
	Display (see page 239)	The display format of the image - horizontal or vertical.
	HeaderBackColor (see page 239)	`HeaderBackColor` sets the header background color.
	HeaderText (see page 239)	The HeaderText of the control.
	PersonalizeExitText (see page 240)	The text of the reset link.
	PersonalizeScopeText (see page 240)	The text of the reset link.
	PersonalizeStartText (see page 240)	The text of the reset link.
	PublicUserButtonText (see page 241)	The text of the reset link.
	ResetLinkText (see page 241)	The text of the reset link.
	ResetLinkToolTip (see page 242)	The text of the reset link tooltip.

Legend

	Property

PersonalizationManager Methods

The methods of the PersonalizationManager class are listed here.

Public Methods

	Name	Description
	Fill (see page 237)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

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 238)	The text of the reset link.
	AddContentText (see page 238)	The text of the reset link.
	CurrentUserButtonText (see page 238)	The text of the reset link.
	Display (see page 239)	The display format of the image - horizontal or vertical.
	HeaderBackColor (see page 239)	`HeaderBackColor` sets the header background color.
	HeaderText (see page 239)	The HeaderText of the control.

	PersonalizeExitText (see page 240)	The text of the reset link.
	PersonalizeScopeText (see page 240)	The text of the reset link.
	PersonalizeStartText (see page 240)	The text of the reset link.
	PublicUserButtonText (see page 241)	The text of the reset link.
	ResetLinkText (see page 241)	The text of the reset link.
	ResetLinkToolTip (see page 242)	The text of the reset link tooltip.

Legend

	Property

PersonalizationManager.AddContentExitText Property

The text of the reset link.

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

The text of the reset link.

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

The text of the reset link.

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_USRSCOPE)]
[Description("The display text for the curront 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 curront user radio button. Default value is set as \""
+ EkWebControl.RES_PMC_USRSCOPE + "\".")>
Public Property CurrentUserButtonText() As string
```

PersonalizationManager.Display Property

The display format of the image - horizontal or 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

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
```

Description

`HeaderBackColor` sets the header background color.

PersonalizationManager.HeaderText Property

The HeaderText of the control.

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

The text of the reset link.

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

The text of the reset link.

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

The text of the reset link.

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

The text of the reset link.

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

The text of the reset link.

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

The text of the reset link tooltip.

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;
```

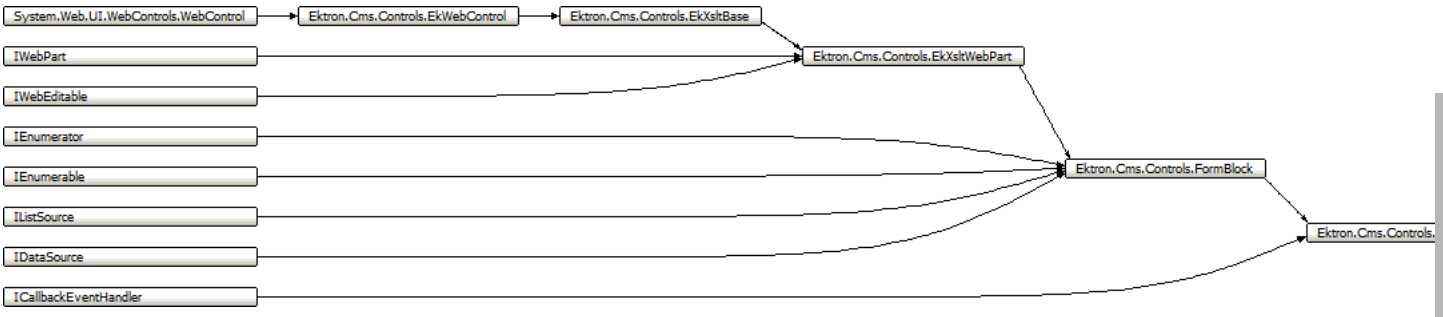
Visual Basic

```
<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
```

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

Members

Poll Methods

	Name	Description
	Fill (see page 243)	Fills in the poll area with the content (see page 337) information from the content (see page 337) database. The form data is retrieved and displayed. Only use if the content (see page 337) must be refreshed after an form ID change or other operation that will require regenerating the display.

Topics

Name	Description
Poll Methods (see page 243)	The methods of the Poll class are listed here.
Poll Properties (see page 244)	The properties of the Poll class are listed here.

Poll Properties

	Name	Description
	EnableAjax (see page 244)	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 (see page 337).

Legend

	Method
	Property

Poll Methods

	Name	Description
	Fill (see page 243)	Fills in the poll area with the content (see page 337) information from the content (see page 337) database. The form data is retrieved and displayed. Only use if the content (see page 337) must be refreshed after an form ID change or other operation that will require regenerating the display.

Legend

	Method

Poll Properties

	Name	Description
	EnableAjax (see page 244)	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 (see page 337).

Legend

	Property

Poll Methods

The methods of the Poll class are listed here.

Public Methods

	Name	Description
	Fill (see page 243)	Fills in the poll area with the content (see page 337) information from the content (see page 337) database. The form data is retrieved and displayed. Only use if the content (see page 337) must be refreshed after an form ID change or other operation that will require regenerating the display.

Legend

	Method

Poll.Fill Method

Fills in the poll area with the content (see page 337) information from the content (see page 337) database. The form data is retrieved and displayed. Only use if the content (see page 337) 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 Properties

The properties of the Poll class are listed here.

Public Properties

	Name	Description
	EnableAjax (see page 244)	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 (see page 337).

Legend

	Property

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 (see page 337).

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(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)>
<Description("Whether the results shown be displayed in the poll location without disturbing
the surrounding contents.")>
Public Property EnableAjax() As bool
```

PostHistory Class

Summary description for Forum.

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

Members

PostHistory Methods

	Name	Description
	Fill (see page 246)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 246)	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 246)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
PostHistory Methods (see page 245)	The methods of the PostHistory class are listed here.
PostHistory Properties (see page 246)	The properties of the PostHistory class are listed here.

PostHistory Properties

	Name	Description
	BoardID (see page 247)	to be changed
	CacheInterval (see page 247)	sets the cache interval in second(s)

Legend

	Method
	Property
	virtual

PostHistory Methods

	Name	Description
	Fill (see page 246)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 246)	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 246)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

PostHistory Properties

	Name	Description
	BoardID (see page 247)	to be changed
	CacheInterval (see page 247)	sets the cache interval in second(s)

Legend

	Property
	virtual

PostHistory Methods

The methods of the PostHistory class are listed here.

Public Methods

	Name	Description
	Fill (see page 246)	Method that can be overridden and used to grab data from the back end for each server control.

	ToString (see page 246)	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 246)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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
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 247)	to be changed
	CacheInterval (see page 247)	sets the cache interval in second(s)

Legend

	Property
	virtual

PostHistory.BoardID Property

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the board you wish to display posts from.")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int BoardID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the board you wish to display posts from.")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BoardID() As Integer
```

Description

to be changed

PostHistory.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)]
public virtual double CacheInterval;
```

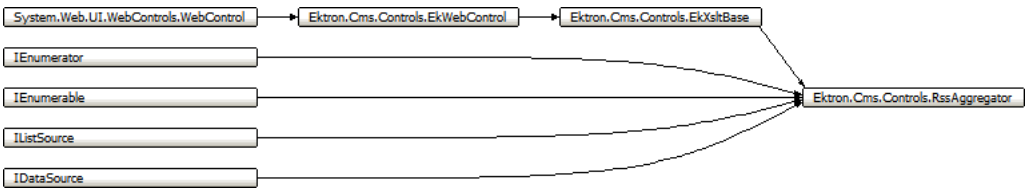
Visual Basic

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

RssAggregator Class

Ektron RssAggregator class. Allows access to a RSS Feed object through IEnumerator, ICollection and IList interfaces.

Class Hierarchy



C#

```
public class RssAggregator : EkXsltBase, IEnumerator, IEnumerable, IListSource, IDataSource;
```

Visual Basic

```
Public Class RssAggregator
Inherits EkXsltBase
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
```

File

RssAggregator.cs

Members

RssAggregator Methods

	Name	Description
	CopyTo (see page 249)	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 250)	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 250)	Gets a System.Collections.IEnumerator for the collection.
	GetView (see page 250)	Gets the view data using dataset
	GetViewNames (see page 250)	gets the view names
	MoveNext (see page 251)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 251)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 251)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.

Topics

Name	Description
RssAggregator Methods (see page 249)	The methods of the RssAggregator class are listed here.
RssAggregator Properties (see page 251)	The properties of the RssAggregator class are listed here.

RssAggregator Properties

	Name	Description
	Count (see page 251)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 252)	Gets the current object in the collection.
	EkItems (see page 252)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	IsSynchronized (see page 252)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 252)	Gets or Sets the target window for the result
	MaxResults (see page 253)	The maximum number of results to fetch. If set to zero, there is no limit.
	SyncRoot (see page 253)	Gets an object that can be used to synchronize the collection.
	URL (see page 253)	Gets or Sets the RSS Feed URL for this Control.

Legend

	Method
	Property

RssAggregator Methods

	Name	Description
	CopyTo (see page 249)	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 250)	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 250)	Gets a System.Collections.IEnumerator for the collection.
	GetView (see page 250)	Gets the view data using dataset
	GetViewNames (see page 250)	gets the view names
	MoveNext (see page 251)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 251)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 251)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.

Legend

	Method

RssAggregator Properties

	Name	Description
	Count (see page 251)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 252)	Gets the current object in the collection.
	EkItems (see page 252)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	IsSynchronized (see page 252)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 252)	Gets or Sets the target window for the result
	MaxResults (see page 253)	The maximum number of results to fetch. If set to zero, there is no limit.
	SyncRoot (see page 253)	Gets an object that can be used to synchronize the collection.
	URL (see page 253)	Gets or Sets the RSS Feed URL for this Control.

Legend

	Property

RssAggregator Methods

The methods of the RssAggregator class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 249)	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 250)	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 250)	Gets a System.Collections.IEnumerator for the collection.
	GetView (see page 250)	Gets the view data using dataset
	GetViewNames (see page 250)	gets the view names
	MoveNext (see page 251)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 251)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 251)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.

Legend

	Method

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
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 63). 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 63) 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 251)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 252)	Gets the current object in the collection.
	EkItems (see page 252)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	IsSynchronized (see page 252)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 252)	Gets or Sets the target window for the result
	MaxResults (see page 253)	The maximum number of results to fetch. If set to zero, there is no limit.
	SyncRoot (see page 253)	Gets an object that can be used to synchronize the collection.
	URL (see page 253)	Gets or Sets the RSS Feed URL for this Control.

Legend

	Property

RssAggregator.Count Property

Indicates the number of elements in your Collection (see page 63).

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 63). 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

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
```

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.")]
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.")>
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.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Gets or Sets the RSS Feed URL for this Control.")]
public string URL;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Gets or Sets the RSS Feed URL for this Control.")>
Public Property URL() As string
```

RssItemBase Class

Container for Title, Link, Description (see page 254) and DatePosted (see page 254) article information.

Class Hierarchy

Ektron.Cms.Controls.RssItemBase

C#

```
public class RssItemBase;
```

Visual Basic

```
Public Class RssItemBase
```

File

RssAggregator.cs

Members

RssItemBase Fields

	Name	Description
	DatePosted (see page 254)	The date the article was posted.
	Description (see page 254)	The description of the article.
	Link (see page 255)	The link for the article.
	Title (see page 255)	The title of the article.

Topics

Name	Description
RssItemBase Fields (see page 254)	The fields of the RssItemBase class are listed here.

Legend

	Data Member

RssItemBase Fields

	Name	Description
	DatePosted (see page 254)	The date the article was posted.
	Description (see page 254)	The description of the article.
	Link (see page 255)	The link for the article.
	Title (see page 255)	The title of the article.

Legend

	Data Member

RssItemBase Fields

The fields of the RssItemBase class are listed here.

Public Fields

	Name	Description
	DatePosted (see page 254)	The date the article was posted.
	Description (see page 254)	The description of the article.
	Link (see page 255)	The link for the article.
	Title (see page 255)	The title of the article.

Legend

	Data Member

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 = "";
```

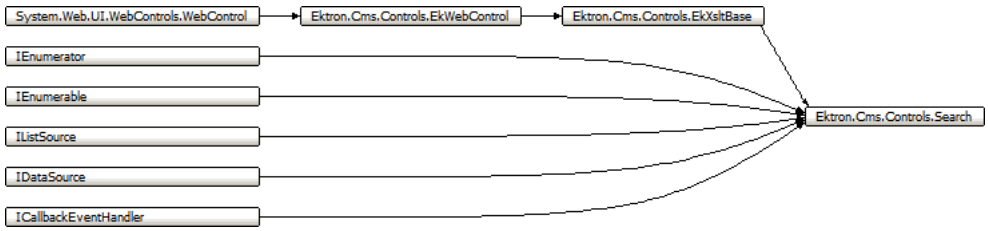
Visual Basic

```
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>")]
public class Search : EkXsltBase, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

Visual Basic

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

File

Search.cs

Members

Public Methods

	Name	Description
	Search (see page 260)	Constructor for Search class

Search Methods

	Name	Description
	CopyTo (see page 261)	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 261)	Method that can be overridden and used to grab data from the back end for each server control.
	GetEnumerator (see page 261)	Gets a System.Collections.IEnumerator for the collection.
	LoadSearch (see page 262)	
	MoveNext (see page 262)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 262)	Brings the index of the collection back to zero. Does not clear the collection.
	SearchDisplayUI (see page 262)	
	this (see page 262)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	Wrap (see page 263)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Topics

Name	Description
Search Methods (see page 261)	The methods of the Search class are listed here.
Search Properties (see page 263)	The properties of the Search class are listed here.

Search Properties

	Name	Description
	AllowFragments (see page 265)	
	ButtonImgSrc (see page 265)	Optional: The path to the image you wish to use for the button.
	ButtonText (see page 265)	The text of the search button if no image is specified.
	ContentFieldsetLegend (see page 266)	The Legend-text to display, for the Content (see page 337)-Group in a fieldset.
	Count (see page 266)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 266)	Gets the current object in the collection.
	Display (see page 266)	The display format of the image - horizontal or vertical.
	EkItems (see page 267)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	EmptyResultMsg (see page 267)	Optional: The return massage if the search returns no hits.
	EnableAdvancedLink (see page 267)	Set to true to display the link that makes the advanced properties visible.
	EnableAjax (see page 267)	Ajax type of search, by default the search is set to PostBack, to enable an Ajax search, set this property to true.
	EnableArchived (see page 268)	Set to true to include Archived items in the search results.
	EnableArchivedBtn (see page 268)	Set to true to display the Archived check box (false to hide, but still use).
	EnableContent (see page 268)	Set to true to include Content (see page 337) items in the search results.
	EnableContentBtn (see page 269)	Set to true to display the Content (see page 337) check box (false to hide, but still use).
	EnableContentFieldset (see page 269)	Set to true to wrap the Content (see page 337)-Group in a fieldset border.
	EnableDMSAssets (see page 269)	Set to true to include DMS-Asset items in the search results.

	EnableDMSAssetsBtn (see page 269)	Set to true to display the DMS-Asset check box (false to hide, but still use).
	EnableFolderFieldset (see page 270)	Set to true to wrap the Folder-Descriptions in a fieldset border.
	EnableForms (see page 270)	Set to true to include Form items in the search results.
	EnableFormsBtn (see page 270)	Set to true to display the Forms check box (false to hide, but still use).
	EnableLibFiles (see page 271)	Set to true to include Library (see page 475)-File items in the search results.
	EnableLibFilesBtn (see page 271)	Set to true to display the Library (see page 475)-Files check box (false to hide, but still use).
	EnableLibHyperlinks (see page 271)	Set to true to include Library (see page 475)-Hyperlink items in the search results.
	EnableLibHyperlinksBtn (see page 271)	Set to true to display the Library (see page 475)-Hyperlinks check box (false to hide, but still use).
	EnableLibImages (see page 272)	Set to true to include Library (see page 475)-Image items in the search results.
	EnableLibImagesBtn (see page 272)	Set to true to display the Library (see page 475)-Images check box (false to hide, but still use).
	EnableLibraryFieldset (see page 272)	Set to true to wrap the Library (see page 475)-Group in a fieldset border.
	FolderFieldsetLegend (see page 273)	The Legend-text to display, for the Folder-Descriptions fieldset.
	FolderID (see page 273)	to be changed
	FolderPath (see page 273)	The Path of the folder you wish to start the search. Root starts at "\".
	IncludeIcons (see page 274)	Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.
	IsSynchronized (see page 274)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LibraryFieldsetLegend (see page 274)	The Legend-text to display, for the Library (see page 475)-Group in a fieldset.
	LinkTarget (see page 275)	Gets or Sets the target window for the result
	MaxCharacters (see page 275)	The maximum number of characters the user can enter in the search box.
	MaxResults (see page 275)	The total number of results to display. If set to zero, unlimited results will be set.
	MaxTeaserLength (see page 275)	An optional result formatting parameter. If non-zero, then the matching content (see page 337) 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 276)	Specify the order of the results. Choices are: • Title - the title of the content (see page 337) • Id - the ID number content (see page 337) • DateCreated - the date the creation date of the content (see page 337) • DateModified - the date the content (see page 337) was last modified • UserId - ID of a user • LastEditorLname - the last name of the last content (see page 337) editor • LastEditorFname - the first name of the last content (see page 337) editor
	OrderDirection (see page 276)	Specify the order direction of the results. Choices are Ascending or Descending.
	PostResultsAt (see page 276)	Specify the search posting page.
	Recursive (see page 277)	Select true if you wish to search the subfolders of the selected Folder.
	RemoveTeaserHtml (see page 277)	An optional result formatting parameter. If true, any HTML in the content (see page 337) 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 277)	Set to true to include Archived items in the search results.
	SearchType (see page 278)	Search Type
	ShowDate (see page 278)	Select true if you wish to display the publication date on each search result.

	ShowExtendedSearch (see page 278)	Set to true to allow a user to search via the metadata type: Searchable Property.
	ShowSearchBoxAlways (see page 278)	If this property is set to false, the search box will not be visible on PostBack.
	ShowSearchOptions (see page 279)	Specify "True" to show search options. Set to false to see a very basic UI.
	Stylesheet (see page 279)	Specify the style sheet relative to site root. Leave blank for default.
	SyncRoot (see page 279)	Gets an object that can be used to synchronize the collection.
	TargetFolderDescriptionList (see page 279)	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 280)	The width of the search box, in characters.
	XmlDoc (see page 280)	Returns the Xslt object for the collection.

Legend

	Constructor
	virtual
	Property

Search Methods

	Name	Description
	CopyTo (see page 261)	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 261)	Method that can be overridden and used to grab data from the back end for each server control.
	GetEnumerator (see page 261)	Gets a System.Collections.IEnumerator for the collection.
	LoadSearch (see page 262)	
	MoveNext (see page 262)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 262)	Brings the index of the collection back to zero. Does not clear the collection.
	SearchDisplayUI (see page 262)	
	this (see page 262)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	Wrap (see page 263)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method
	virtual

Search Properties

	Name	Description
	AllowFragments (see page 265)	
	ButtonImgSrc (see page 265)	Optional: The path to the image you wish to use for the button.
	ButtonText (see page 265)	The text of the search button if no image is specified.
	ContentFieldsetLegend (see page 266)	The Legend-text to display, for the Content (see page 337)-Group in a fieldset.
	Count (see page 266)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 266)	Gets the current object in the collection.
	Display (see page 266)	The display format of the image - horizontal or vertical.
	EkItems (see page 267)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	EmptyResultMsg (see page 267)	Optional: The return massage if the search returns no hits.
	EnableAdvancedLink (see page 267)	Set to true to display the link that makes the advanced properties visible.
	EnableAjax (see page 267)	Ajax type of search, by default the search is set to PostBack, to enable an Ajax search, set this property to true.

	EnableArchived (see page 268)	Set to true to include Archived items in the search results.
	EnableArchivedBtn (see page 268)	Set to true to display the Archived check box (false to hide, but still use).
	EnableContent (see page 268)	Set to true to include Content (see page 337) items in the search results.
	EnableContentBtn (see page 269)	Set to true to display the Content (see page 337) check box (false to hide, but still use).
	EnableContentFieldset (see page 269)	Set to true to wrap the Content (see page 337)-Group in a fieldset border.
	EnableDMSAssets (see page 269)	Set to true to include DMS-Asset items in the search results.
	EnableDMSAssetsBtn (see page 269)	Set to true to display the DMS-Asset check box (false to hide, but still use).
	EnableFolderFieldset (see page 270)	Set to true to wrap the Folder-Descriptions in a fieldset border.
	EnableForms (see page 270)	Set to true to include Form items in the search results.
	EnableFormsBtn (see page 270)	Set to true to display the Forms check box (false to hide, but still use).
	EnableLibFiles (see page 271)	Set to true to include Library (see page 475)-File items in the search results.
	EnableLibFilesBtn (see page 271)	Set to true to display the Library (see page 475)-Files check box (false to hide, but still use).
	EnableLibHyperlinks (see page 271)	Set to true to include Library (see page 475)-Hyperlink items in the search results.
	EnableLibHyperlinksBtn (see page 271)	Set to true to display the Library (see page 475)-Hyperlinks check box (false to hide, but still use).
	EnableLibImages (see page 272)	Set to true to include Library (see page 475)-Image items in the search results.
	EnableLibImagesBtn (see page 272)	Set to true to display the Library (see page 475)-Images check box (false to hide, but still use).
	EnableLibraryFieldset (see page 272)	Set to true to wrap the Library (see page 475)-Group in a fieldset border.
	FolderFieldsetLegend (see page 273)	The Legend-text to display, for the Folder-Descriptions fieldset.
	FolderID (see page 273)	to be changed
	FolderPath (see page 273)	The Path of the folder you wish to start the search. Root starts at "\".
	IncludeIcons (see page 274)	Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.
	IsSynchronized (see page 274)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LibraryFieldsetLegend (see page 274)	The Legend-text to display, for the Library (see page 475)-Group in a fieldset.
	LinkTarget (see page 275)	Gets or Sets the target window for the result
	MaxCharacters (see page 275)	The maximum number of characters the user can enter in the search box.
	MaxResults (see page 275)	The total number of results to display. If set to zero, unlimited results will be set.
	MaxTeaserLength (see page 275)	An optional result formatting parameter. If non-zero, then the matching content (see page 337) 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 276)	Specify the order of the results. Choices are: • Title - the title of the content (see page 337) • Id - the ID number content (see page 337) • DateCreated - the date the creation date of the content (see page 337) • DateModified - the date the content (see page 337) was last modified • UserId - ID of a user • LastEditorLname - the last name of the last content (see page 337) editor • LastEditorFname - the first name of the last content (see page 337) editor

	OrderDirection (see page 276)	Specify the order direction of the results. Choices are Ascending or Descending.
	PostResultsAt (see page 276)	Specify the search posting page.
	Recursive (see page 277)	Select true if you wish to search the subfolders of the selected Folder.
	RemoveTeaserHtml (see page 277)	An optional result formatting parameter. If true, any HTML in the content (see page 337) 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 277)	Set to true to include Archived items in the search results.
	SearchType (see page 278)	Search Type
	ShowDate (see page 278)	Select true if you wish to display the publication date on each search result.
	ShowExtendedSearch (see page 278)	Set to true to allow a user to search via the metadata type: Searchable Property.
	ShowSearchBoxAlways (see page 278)	If this property is set to false, the search box will not be visible on PostBack.
	ShowSearchOptions (see page 279)	Specify "True" to show search options. Set to false to see a very basic UI.
	Stylesheet (see page 279)	Specify the style sheet relative to site root. Leave blank for default.
	SyncRoot (see page 279)	Gets an object that can be used to synchronize the collection.
	TargetFolderDescriptionList (see page 279)	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 280)	The width of the search box, in characters.
	XmlDoc (see page 280)	Returns the Xslt object for the collection.

Legend

	Property

Public Methods

	Name	Description
	Search (see page 260)	Constructor for Search class

Legend

	Constructor

Search.Search Constructor

Constructor for Search class

C#

```
public Search();
```

Visual Basic

```
Public Sub Search()
```

Search Methods

The methods of the Search class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 261)	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 261)	Method that can be overridden and used to grab data from the back end for each server control.
	GetEnumerator (see page 261)	Gets a System.Collections.IEnumerator for the collection.
	LoadSearch (see page 262)	
	MoveNext (see page 262)	Moves up the index of the Collection (see page 63). Returns false if the end of the collection has been surpassed.
	Reset (see page 262)	Brings the index of the collection back to zero. Does not clear the collection.

	SearchDisplayUI (see page 262)	
	this (see page 262)	Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) item.
	Wrap (see page 263)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method
	virtual

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
SearchReq	

Search.MoveNext Method

Moves up the index of the Collection (see page 63). 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
supressPostRedirect	

Search.this Indexer

Index operator into a strongly typed dataset holding some common properties for a Collection (see page 63) 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
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
	AllowFragments (see page 265)	
	ButtonImgSrc (see page 265)	Optional: The path to the image you wish to use for the button.
	ButtonText (see page 265)	The text of the search button if no image is specified.
	ContentFieldsetLegend (see page 266)	The Legend-text to display, for the Content (see page 337)-Group in a fieldset.
	Count (see page 266)	Indicates the number of elements in your Collection (see page 63).
	Current (see page 266)	Gets the current object in the collection.
	Display (see page 266)	The display format of the image - horizontal or vertical.
	EkItems (see page 267)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 63). Holds more information than the exposed strongly typed this[i] index operator.
	EmptyResultMsg (see page 267)	Optional: The return massage if the search returns no hits.
	EnableAdvancedLink (see page 267)	Set to true to display the link that makes the advanced properties visible.
	EnableAjax (see page 267)	Ajax type of search, by default the search is set to PostBack, to enable an Ajax search, set this property to true.
	EnableArchived (see page 268)	Set to true to include Archived items in the search results.
	EnableArchivedBtn (see page 268)	Set to true to display the Archived check box (false to hide, but still use).
	EnableContent (see page 268)	Set to true to include Content (see page 337) items in the search results.
	EnableContentBtn (see page 269)	Set to true to display the Content (see page 337) check box (false to hide, but still use).
	EnableContentFieldset (see page 269)	Set to true to wrap the Content (see page 337)-Group in a fieldset border.
	EnableDMSAssets (see page 269)	Set to true to include DMS-Asset items in the search results.
	EnableDMSAssetsBtn (see page 269)	Set to true to display the DMS-Asset check box (false to hide, but still use).
	EnableFolderFieldset (see page 270)	Set to true to wrap the Folder-Descriptions in a fieldset border.
	EnableForms (see page 270)	Set to true to include Form items in the search results.
	EnableFormsBtn (see page 270)	Set to true to display the Forms check box (false to hide, but still use).
	EnableLibFiles (see page 271)	Set to true to include Library (see page 475)-File items in the search results.
	EnableLibFilesBtn (see page 271)	Set to true to display the Library (see page 475)-Files check box (false to hide, but still use).
	EnableLibHyperlinks (see page 271)	Set to true to include Library (see page 475)-Hyperlink items in the search results.
	EnableLibHyperlinksBtn (see page 271)	Set to true to display the Library (see page 475)-Hyperlinks check box (false to hide, but still use).
	EnableLibImages (see page 272)	Set to true to include Library (see page 475)-Image items in the search results.
	EnableLibImagesBtn (see page 272)	Set to true to display the Library (see page 475)-Images check box (false to hide, but still use).
	EnableLibraryFieldset (see page 272)	Set to true to wrap the Library (see page 475)-Group in a fieldset border.
	FolderFieldsetLegend (see page 273)	The Legend-text to display, for the Folder-Descriptions fieldset.
	FolderID (see page 273)	to be changed
	FolderPath (see page 273)	The Path of the folder you wish to start the search. Root starts at "\".

	IncludeIcons (see page 274)	Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.
	IsSynchronized (see page 274)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LibraryFieldsetLegend (see page 274)	The Legend-text to display, for the Library (see page 475)-Group in a fieldset.
	LinkTarget (see page 275)	Gets or Sets the target window for the result
	MaxCharacters (see page 275)	The maximum number of characters the user can enter in the search box.
	MaxResults (see page 275)	The total number of results to display. If set to zero, unlimited results will be set.
	MaxTeaserLength (see page 275)	An optional result formatting parameter. If non-zero, then the matching content (see page 337) 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 276)	Specify the order of the results. Choices are: • Title - the title of the content (see page 337) • Id - the ID number content (see page 337) • DateCreated - the date the creation date of the content (see page 337) • DateModified - the date the content (see page 337) was last modified • UserId - ID of a user • LastEditorLname - the last name of the last content (see page 337) editor • LastEditorFname - the first name of the last content (see page 337) editor
	OrderDirection (see page 276)	Specify the order direction of the results. Choices are Ascending or Descending.
	PostResultsAt (see page 276)	Specify the search posting page.
	Recursive (see page 277)	Select true if you wish to search the subfolders of the selected Folder.
	RemoveTeaserHtml (see page 277)	An optional result formatting parameter. If true, any HTML in the content (see page 337) 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 277)	Set to true to include Archived items in the search results.
	SearchType (see page 278)	Search Type
	ShowDate (see page 278)	Select true if you wish to display the publication date on each search result.
	ShowExtendedSearch (see page 278)	Set to true to allow a user to search via the metadata type: Searchable Property.
	ShowSearchBoxAlways (see page 278)	If this property is set to false, the search box will not be visible on PostBack.
	ShowSearchOptions (see page 279)	Specify "True" to show search options. Set to false to see a very basic UI.
	Stylesheet (see page 279)	Specify the style sheet relative to site root. Leave blank for default.
	SyncRoot (see page 279)	Gets an object that can be used to synchronize the collection.
	TargetFolderDescriptionList (see page 279)	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 280)	The width of the search box, in characters.
	XmlDoc (see page 280)	Returns the Xslt object for the collection.

Legend

	Property

Search.AllowFragments Property

C#

```
[Browsable(false)]
public bool AllowFragments;
```

Visual Basic

```
<Browsable(false)>
Public Property AllowFragments() As bool
```

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

The Legend-text to display, for the Content (see page 337)-Group in a fieldset.

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 your Collection (see page 63).

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
```

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

The display format of the image - horizontal or 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 63). 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

Ajax type of search, by default the search is set to PostBack, to enable an Ajax search, set this property to true.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Enables Ajax type of search")]
public bool EnableAjax;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Enables Ajax type of search")>
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 display the Archived check box (false to hide, but still use).

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 337) 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 display the Content (see page 337) check box (false to hide, but still use).

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

Set to true to wrap the Content (see page 337)-Group in a fieldset border.

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 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 display the DMS-Asset check box (false to hide, but still use).

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-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 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 display the Forms check box (false to hide, but still use).

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 475)-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 display the Library (see page 475)-Files check box (false to hide, but still use).

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 475)-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 display the Library (see page 475)-Hyperlinks check box (false to hide, but still use).

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 475)-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 475)-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 475)-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-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

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))]
public int 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))>
Public Property FolderID() As Integer
```

Description

to be changed

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.

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 475)-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

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 (see page 337) 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 (see page 337)
- Id - the ID number content (see page 337)

- DateCreated - the date the creation date of the content (see page 337)
- DateModified - the date the content (see page 337) was last modified
- UserId - ID of a user
- LastEditorLname - the last name of the last content (see page 337) editor
- LastEditorFname - the first name of the last content (see page 337) 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

Specify the search 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

Select true if you wish to search the subfolders of the selected Folder.

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 (see page 337) 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

Set to true to include Archived items in the search results.

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.SearchType Property

Search 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

Set to true to allow a user to search via the metadata type: Searchable Property.

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
```

Search.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
```

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

This control display the sitemap from the folder breadcrumbs.

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

Members

Sitemap Methods

	Name	Description
	Fill (see page 282)	Method that can be overridden and used to grab data from the back end for each server control.

Topics

Name	Description
Sitemap Methods (see page 281)	The methods of the Sitemap class are listed here.
Sitemap Properties (see page 282)	The properties of the Sitemap class are listed here.

Sitemap Properties

	Name	Description
	CacheInterval (see page 282)	Sets the cache interval in second(s).
	ClassName (see page 282)	Stylesheet class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.
	DisplayType (see page 283)	Display type controls how sitemap get render on the page.
	FlatTopLevel (see page 283)	Includes site nodes from the parent folder but not thier items.
	FolderID (see page 284)	to be changed
	MaxLevel (see page 284)	Set max level depth of the sitemap.
	StartingLevel (see page 284)	Set max level depth of the sitemap.
	XmlDoc (see page 285)	Returns the XML object for the sitemap.

Legend

	Method
	Property
	virtual

Sitemap Methods

	Name	Description
	Fill (see page 282)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

Sitemap Properties

	Name	Description
	CacheInterval (see page 282)	Sets the cache interval in second(s).
	ClassName (see page 282)	Stylesheet class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.
	DisplayType (see page 283)	Display type controls how sitemap get render on the page.
	FlatTopLevel (see page 283)	Includes site nodes from the parent folder but not thier items.
	FolderID (see page 284)	to be changed
	MaxLevel (see page 284)	Set max level depth of the sitemap.
	StartingLevel (see page 284)	Set max level depth of the sitemap.
	XmlDoc (see page 285)	Returns the XML object for the sitemap.

Legend

	Property
	virtual

Sitemap Methods

The methods of the Sitemap class are listed here.

Public Methods

	Name	Description
	Fill (see page 282)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

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 282)	Sets the cache interval in second(s).
	ClassName (see page 282)	Stylesheet class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.
	DisplayType (see page 283)	Display type controls how sitemap get render on the page.
	FlatTopLevel (see page 283)	Includes site nodes from the parent folder but not thier items.
	FolderID (see page 284)	to be changed
	MaxLevel (see page 284)	Set max level depth of the sitemap.
	StartingLevel (see page 284)	Set max level depth of the sitemap.
	XmlDoc (see page 285)	Returns the XML object for the sitemap.

Legend

	Property
	virtual

Sitemap.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)]
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
```

Sitemap.ClassName Property

Stylesheet 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.\nLeave 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.\nLeave blank to accept default.")>
Public Property ClassName() As string
```

Sitemap.DisplayType Property

Display type controls how sitemap get render on the page.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Browsable(true)]
[Description("Display type controls how sitemap get render 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 get render on the page.")>
<Editor(typeof(SitemapDisplayTypeConverter), typeof(System.Drawing.Design.UITypeEditor))>
Public Property DisplayType() As string
```

Sitemap.FlatTopLevel Property

Includes site nodes from the parent folder but not thier items.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Includes site nodes from the parent folder but not thier items")]
[DefaultValue(false)]
public bool FlatTopLevel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Includes site nodes from the parent folder but not thier items")>
<DefaultValue(false)>
Public Property FlatTopLevel() As bool
```

Description

.

Sitemap.FolderID Property

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))]
public int 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))>
Public Property FolderID() As Integer
```

Description

to be changed

Sitemap.MaxLevel Property

Set max level depth of the sitemap.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("Set the max level of the sitemap. Set it to 0 for unlimited.")]
public int MaxLevel;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("Set the max level of the sitemap. Set it to 0 for unlimited.")>
Public Property MaxLevel() As Integer
```

Sitemap.StartingLevel Property

Set max level depth of the sitemap.

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.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-Sever-Control, adding CSS based menus.

Class Hierarchy

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

Members

SmartMenu Properties

	Name	Description
	AutoCollapseBranches (see page 287)	Set to automatically close other menu branches when a new one is opened
	DisplayXslt (see page 287)	The Xslt to use to render the Menu (optional)
	Enable508Compliance (see page 287)	Set to enable 508 Compliance (make menus accessible to disabled users)
	Enable508NavigationHeadings (see page 288)	Set to enable accessibility (508-Compliant) navigation headings (only used if Enable508Compliance (see page 287) is true)
	EnableMouseOverPopUp (see page 288)	Set to enable submenu activation (pop-ups) on mouse-over.
	EnableSmartOpen (see page 288)	When true, menu will auto-open to show matching IDs or folder associations.
	EnableSubmenuLinkPathMatching (see page 289)	Set to enable auto-submenu opening based on link match (if all else fails).
	EnableSubmenuTemplateMatching (see page 289)	Set to enable auto-submenu opening based on a template filename match (if all else fails).
	LaunchLinksInNewWindow (see page 289)	Set to launch external links in a new window
	ShowRootFolder (see page 290)	Set to make the root menu folder-button visible (normally not desired)
	SingleEditButton (see page 290)	Set to only render a single edit button (and no add button) when logged in.
	StartCollapsed (see page 290)	Set to initially render menu with submenus closed
	StartWithRootCollapsed (see page 291)	Set to initially render menu with root-menu open: it's contents made visible
	Stylesheet (see page 291)	The Style-Sheet to use to render the Menu (optional)

Topics

Name	Description
SmartMenu Properties (see page 286)	The properties of the SmartMenu class are listed here.

Legend

	Property

SmartMenu Properties

	Name	Description
	AutoCollapseBranches (see page 287)	Set to automatically close other menu branches when a new one is opened

	DisplayXslt (see page 287)	The Xslt to use to render the Menu (optional)
	Enable508Compliance (see page 287)	Set to enable 508 Compliance (make menus accessible to disabled users)
	Enable508NavigationHeadings (see page 288)	Set to enable accessibility (508-Compliant) navigation headings (only used if Enable508Compliance (see page 287) is true)
	EnableMouseOverPopUp (see page 288)	Set to enable submenu activation (pop-ups) on mouse-over.
	EnableSmartOpen (see page 288)	When true, menu will auto-open to show matching IDs or folder associations.
	EnableSubmenuLinkPathMatching (see page 289)	Set to enable auto-submenu opening based on link match (if all else fails).
	EnableSubmenuTemplateMatching (see page 289)	Set to enable auto-submenu opening based on a template filename match (if all else fails).
	LaunchLinksInNewWindow (see page 289)	Set to launch external links in a new window
	ShowRootFolder (see page 290)	Set to make the root menu folder-button visible (normally not desired)
	SingleEditButton (see page 290)	Set to only render a single edit button (and no add button) when logged in.
	StartCollapsed (see page 290)	Set to initially render menu with submenus closed
	StartWithRootCollapsed (see page 291)	Set to initially render menu with root-menu open: it's contents made visible
	Stylesheet (see page 291)	The Style-Sheet to use to render the Menu (optional)

Legend

	Property

SmartMenu Properties

The properties of the SmartMenu class are listed here.

Public Properties

	Name	Description
	AutoCollapseBranches (see page 287)	Set to automatically close other menu branches when a new one is opened
	DisplayXslt (see page 287)	The Xslt to use to render the Menu (optional)
	Enable508Compliance (see page 287)	Set to enable 508 Compliance (make menus accessible to disabled users)
	Enable508NavigationHeadings (see page 288)	Set to enable accessibility (508-Compliant) navigation headings (only used if Enable508Compliance (see page 287) is true)
	EnableMouseOverPopUp (see page 288)	Set to enable submenu activation (pop-ups) on mouse-over.
	EnableSmartOpen (see page 288)	When true, menu will auto-open to show matching IDs or folder associations.
	EnableSubmenuLinkPathMatching (see page 289)	Set to enable auto-submenu opening based on link match (if all else fails).
	EnableSubmenuTemplateMatching (see page 289)	Set to enable auto-submenu opening based on a template filename match (if all else fails).
	LaunchLinksInNewWindow (see page 289)	Set to launch external links in a new window
	ShowRootFolder (see page 290)	Set to make the root menu folder-button visible (normally not desired)
	SingleEditButton (see page 290)	Set to only render a single edit button (and no add button) when logged in.
	StartCollapsed (see page 290)	Set to initially render menu with submenus closed
	StartWithRootCollapsed (see page 291)	Set to initially render menu with root-menu open: it's contents made visible
	Stylesheet (see page 291)	The Style-Sheet to use to render the Menu (optional)

Legend

	Property

SmartMenu.AutoCollapseBranches Property

Set to automatically close other menu branches when a new one is opened

C#

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

Set to enable 508 Compliance (make menus accessible to disabled users)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set 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 enable 508 Compliance (make menus accessible to disabled users)")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property Enable508Compliance() As bool
```

SmartMenu.Enable508NavigationHeadings Property

Set to enable accessibility (508-Compliant) navigation headings (only used if Enable508Compliance (see page 287) is true)

C#

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

Set to enable submenu activation (pop-ups) on mouse-over.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set 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 enable submenu activation (pop-ups) on mouse-over.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property EnableMouseOverPopUp() As bool
```

SmartMenu.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
```

SmartMenu.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
```

SmartMenu.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
```

SmartMenu.LaunchLinksInNewWindow Property

Set to launch external links in a new window

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set 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 launch external links in a new window")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property LaunchLinksInNewWindow() As bool
```

SmartMenu.ShowRootFolder Property

Set to make the root menu folder-button visible (normally not desired)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set 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 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 only render a single edit button (and no add button) when logged in.

C#

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

Set 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
```

SmartMenu.StartWithRootCollapsed Property

Set to initially render menu with root-menu open: it's contents made visible

C#

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

The Style-Sheet to use to render the Menu (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
```

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 exposes 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 292)	This is namespace Ektron.Cms.Extensibility.

Ektron.Cms.Extensibility Namespace

This is namespace Ektron.Cms.Extensibility.

Classes

	Name	Description
	ContentInfo (see page 293)	Class which holds information about the content (see page 337) 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.
	ExtensionAuthor (see page 297)	Custom attribute - Used at the method and/or assembly level to indicate who wrote the plug-in.
	ExtensionDescription (see page 298)	Custom attribute - Used at the method level to describe what a particular event does
	ExtensionEvent (see page 298)	Override the methods in this class to customize the CMS via event hooks.
	FolderInfo (see page 316)	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 319)	Class that describes a field in a form - name / value pair
	FormInfo (see page 320)	Class which describes a Form object
	SiteInfo (see page 323)	Class which holds information about the site in which the event was fired.
	UserInfo (see page 325)	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.
	WebServiceProxy (see page 328)	This is class Ektron.Cms.Extensibility.WebServiceProxy.

Legend

	Class
	abstract

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	ContentInfo (see page 293)	Class which holds information about the content (see page 337) 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.
	ExtensionAuthor (see page 297)	Custom attribute - Used at the method and/or assembly level to indicate who wrote the plug-in.
	ExtensionDescription (see page 298)	Custom attribute - Used at the method level to describe what a particular event does
	ExtensionEvent (see page 298)	Override the methods in this class to customize the CMS via event hooks.
	FolderInfo (see page 316)	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 319)	Class that describes a field in a form - name / value pair
	FormInfo (see page 320)	Class which describes a Form object
	SiteInfo (see page 323)	Class which holds information about the site in which the event was fired.
	UserInfo (see page 325)	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.
	WebServiceProxy (see page 328)	This is class Ektron.Cms.Extensibility.WebServiceProxy.

Legend

	Class
	abstract

ContentInfo Class

Class which holds information about the content (see page 337) 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

C#

```
public sealed class ContentInfo;
```

Visual Basic

```
Public sealed Class ContentInfo
```

File

CmsInfoObjects.cs

Members

ContentInfo Properties

	Name	Description
	AssetID (see page 295)	Gets a string containing the asset ID in the form of a GUID if the content (see page 337) is an asset. (ReadOnly)
	Comments (see page 295)	Gets a string containing the comments a user has entered during their operation. (ReadOnly)
	ErrorMessage (see page 295)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	FileAndPath (see page 295)	Gets a string containing the full physical path and file name on the server if content (see page 337) is an asset. (ReadOnly)
	FileExtension (see page 295)	Gets a string containing file extension if content (see page 337) is an asset. (ReadOnly)
	FileName (see page 296)	Gets a string containing file name if content (see page 337) is an asset. (ReadOnly)
	FolderID (see page 296)	Gets an integer representing the folder ID which this content (see page 337) is located. (ReadOnly)
	Html (see page 296)	Gets or Sets a string containing all the html displayed to end users for a piece of content (see page 337). (ReadWrite Note: This property only allows write access within OnBeforePublish.)
	ID (see page 296)	Gets an integer representing the content (see page 337) ID of the content (see page 337). (ReadOnly)
	Language (see page 296)	Gets an integer representing the language of the content (see page 337). (ReadOnly)
	MimeType (see page 296)	Gets a string containing the MIME type if content (see page 337) is an asset. (ReadOnly)
	Status (see page 297)	Gets a string representing the status of the content (see page 337), published, checkedOut. (ReadOnly)
	Title (see page 297)	Gets a string containing the title of the content (see page 337). (ReadOnly)
	Type (see page 297)	Gets an integer representing the content (see page 337) type. (ReadOnly)
	UserID (see page 297)	Gets an integer representing the user ID of the user performing the operation on the content (see page 337). (ReadOnly)

Topics

Name	Description
ContentInfo Properties (see page 294)	The properties of the ContentInfo class are listed here.

Legend

	Property

ContentInfo Properties

	Name	Description
	AssetID (see page 295)	Gets a string containing the asset ID in the form of a GUID if the content (see page 337) is an asset. (ReadOnly)
	Comments (see page 295)	Gets a string containing the comments a user has entered during their operation. (ReadOnly)

	ErrorMessage (see page 295)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	FileAndPath (see page 295)	Gets a string containing the full physical path and file name on the server if content (see page 337) is an asset. (ReadOnly)
	FileExtension (see page 295)	Gets a string containing file extension if content (see page 337) is an asset. (ReadOnly)
	FileName (see page 296)	Gets a string containing file name if content (see page 337) is an asset. (ReadOnly)
	FolderID (see page 296)	Gets an integer representing the folder ID which this content (see page 337) is located. (ReadOnly)
	Html (see page 296)	Gets or Sets a string containing all the html displayed to end users for a piece of content (see page 337). (ReadWrite Note: This property only allows write access within OnBeforePublish.)
	ID (see page 296)	Gets an integer representing the content (see page 337) ID of the content (see page 337). (ReadOnly)
	Language (see page 296)	Gets an integer representing the language of the content (see page 337). (ReadOnly)
	MimeType (see page 296)	Gets a string containing the MIME type if content (see page 337) is an asset. (ReadOnly)
	Status (see page 297)	Gets a string representing the status of the content (see page 337), published, checkedOut. (ReadOnly)
	Title (see page 297)	Gets a string containing the title of the content (see page 337). (ReadOnly)
	Type (see page 297)	Gets an integer representing the content (see page 337) type. (ReadOnly)
	UserID (see page 297)	Gets an integer representing the user ID of the user performing the operation on the content (see page 337). (ReadOnly)

Legend

	Property

ContentInfo Properties

The properties of the ContentInfo class are listed here.

Public Properties

	Name	Description
	AssetID (see page 295)	Gets a string containing the asset ID in the form of a GUID if the content (see page 337) is an asset. (ReadOnly)
	Comments (see page 295)	Gets a string containing the comments a user has entered during their operation. (ReadOnly)
	ErrorMessage (see page 295)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	FileAndPath (see page 295)	Gets a string containing the full physical path and file name on the server if content (see page 337) is an asset. (ReadOnly)
	FileExtension (see page 295)	Gets a string containing file extension if content (see page 337) is an asset. (ReadOnly)
	FileName (see page 296)	Gets a string containing file name if content (see page 337) is an asset. (ReadOnly)
	FolderID (see page 296)	Gets an integer representing the folder ID which this content (see page 337) is located. (ReadOnly)
	Html (see page 296)	Gets or Sets a string containing all the html displayed to end users for a piece of content (see page 337). (ReadWrite Note: This property only allows write access within OnBeforePublish.)
	ID (see page 296)	Gets an integer representing the content (see page 337) ID of the content (see page 337). (ReadOnly)
	Language (see page 296)	Gets an integer representing the language of the content (see page 337). (ReadOnly)
	MimeType (see page 296)	Gets a string containing the MIME type if content (see page 337) is an asset. (ReadOnly)
	Status (see page 297)	Gets a string representing the status of the content (see page 337), published, checkedOut. (ReadOnly)
	Title (see page 297)	Gets a string containing the title of the content (see page 337). (ReadOnly)
	Type (see page 297)	Gets an integer representing the content (see page 337) type. (ReadOnly)
	UserID (see page 297)	Gets an integer representing the user ID of the user performing the operation on the content (see page 337). (ReadOnly)

Legend

	Property

ContentInfo.AssetID Property

Gets a string containing the asset ID in the form of a GUID if the content (see page 337) 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 (see page 337) 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 (see page 337) 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 (see page 337) is an asset. (ReadOnly)

C#

```
public string FileName;
```

Visual Basic

```
Public Property FileName() As string
```

ContentInfo.FolderID Property

Gets an integer representing the folder ID which this content (see page 337) is located. (ReadOnly)

C#

```
public int FolderID;
```

Visual Basic

```
Public Property FolderID() As Integer
```

ContentInfo.Html Property

Gets or Sets a string containing all the html displayed to end users for a piece of content (see page 337). (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 integer representing the content (see page 337) ID of the content (see page 337). (ReadOnly)

C#

```
public int ID;
```

Visual Basic

```
Public Property ID() As Integer
```

ContentInfo.Language Property

Gets an integer representing the language of the content (see page 337). (ReadOnly)

C#

```
public int Language;
```

Visual Basic

```
Public Property Language() As Integer
```

ContentInfo.MimeType Property

Gets a string containing the MIME type if content (see page 337) 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 (see page 337), 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 (see page 337). (ReadOnly)

C#

```
public string Title;
```

Visual Basic

```
Public Property Title() As string
```

ContentInfo.Type Property

Gets an integer representing the content (see page 337) type. (ReadOnly)

C#

```
public int Type;
```

Visual Basic

```
Public Property Type() As Integer
```

ContentInfo.UserID Property

Gets an integer representing the user ID of the user performing the operation on the content (see page 337). (ReadOnly)

C#

```
public int UserID;
```

Visual Basic

```
Public Property UserID() As Integer
```

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
```

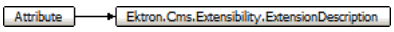
File

ExtensibilityBase.cs

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

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

Members

ExtensionEvent Methods

	Name	Description
	GetActiveDirectoryUserProxy (see page 304)	Returns an ActiveDirectoryUserProxy object to access the ActiveDirectoryUser (see page 545) Web Service.
	GetAssetProxy (see page 304)	Returns an AssetProxy object to access the Asset Web Service.
	GetBlogProxy (see page 304)	Returns a BlogProxy object to access the Blog Web Service.
	GetBusinessRulesProxy (see page 305)	Returns a BusinessRulesProxy object to access the BusinessRules (see page 54) Web Service.
	GetCalendarEventTypeProxy (see page 305)	Returns a CalendarEventTypeProxy object to access the CalendarEventType (see page 337) Web Service.
	GetCalendarProxy (see page 305)	Returns a CalendarProxy object to access the Calendar (see page 336) Web Service.
	GetContentProxy (see page 305)	Returns a ContentProxy object to access the Content (see page 315) Web Service.
	GetContentRatingProxy (see page 306)	Returns a ContentRatingProxy object to access the ContentRating (see page 379) Web Service.
	GetCustomFieldsProxy (see page 306)	Returns a CustomFieldsProxy object to access the CustomFields (see page 446) Web Service.
	GetFolderProxy (see page 306)	Returns a FolderProxy object to access the Folder Web Service.
	GetFontProxy (see page 306)	Returns a FontProxy object to access the Font Web Service.
	GetFormProxy (see page 306)	Returns a FormProxy object to access the Form Web Service.
	GetLibraryProxy (see page 307)	Returns a LibraryProxy object to access the Library (see page 475) Web Service.
	GetMetadataProxy (see page 307)	Returns a MetadataProxy object to access the Metadata (see page 495) Web Service.

	GetObjectStateAsXml (see page 307)	Retrieves changes made by individual plug-ins
	GetPermissionsProxy (see page 307)	Returns a PermissionsProxy object to access the Permissions (see page 500) Web Service.
	GetSearchManagerProxy (see page 308)	Returns a SearchManagerProxy object to access the SearchManager (see page 400) Web Service.
	GetSiteMapProxy (see page 308)	Returns a SiteMapProxy object to access the SiteMap (see page 512) Web Service.
	GetSiteProxy (see page 308)	Returns a SiteProxy object to access the Site Web Service.
	GetTaskCategoryProxy (see page 308)	Returns a TaskCategoryProxy object to access the Task Category Web Service.
	GetTaskCategoryTypeProxy (see page 308)	Returns a TaskProxy object to access the Task Web Service.
	GetTaskProxy (see page 309)	Returns a TaskProxy object to access the Task Web Service.
	GetThreadedDiscussionProxy (see page 309)	Returns a ThreadedDiscussionProxy object to access the ThreadedDiscussion (see page 390) Web Service.
	GetUserProxy (see page 309)	Returns a UserProxy object to access the User Web Service.
	OnAfterContentAdd (see page 309)	This hook is called after content (see page 337) is added in the CMS.
	OnAfterContentDelete (see page 310)	This hook is called after a delete of content (see page 337) is completed in the CMS.
	OnAfterContentEdit (see page 310)	This hook is called after an edit is completed in the CMS.
	OnAfterFolderAdd (see page 310)	This hook is called after a folder is created in the CMS.
	OnAfterFolderDelete (see page 310)	This hook is called after a folder is deleted in the CMS.
	OnAfterFolderEdit (see page 311)	This hook is called after a folder edit is completed in the CMS.
	OnAfterFormSubmit (see page 311)	This hook is called after form data is submitted to the CMS database.
	OnAfterPublish (see page 311)	This hook is called after a publish is completed in the CMS.
	OnAfterUserAdd (see page 311)	This hook is called after a user is created in the CMS.
	OnAfterUserDelete (see page 312)	This hook is called after a user is deleted from the CMS.
	OnAfterUserEdit (see page 312)	This hook is called after a user is edited in the CMS.
	OnAfterUserLogin (see page 312)	This hook is called after a user logs into in the CMS.
	OnBeforeContentAdd (see page 312)	This hook is called before content (see page 337) is added to the CMS.
	OnBeforeContentDelete (see page 312)	This hook is called before a delete of content (see page 337) in the CMS.
	OnBeforeContentEdit (see page 313)	This hook is called before a content (see page 337) is edited in the CMS.
	OnBeforeFolderAdd (see page 313)	This hook is called before a folder is created in the CMS.
	OnBeforeFolderDelete (see page 313)	This hook is called before a folder is deleted from the CMS.
	OnBeforeFolderEdit (see page 313)	This hook is called before a folder is edited in the CMS.
	OnBeforeFormSubmit (see page 314)	This hook is called before form data is submitted to the CMS database.
	OnBeforePublish (see page 314)	This hook is called before a publish takes place in the CMS.
	OnBeforeUserAdd (see page 314)	This hook is called before a user is created in the CMS.
	OnBeforeUserDelete (see page 314)	This hook is called before a user is delete from the CMS.

	OnBeforeUserEdit (see page 315)	This hook is called before a user is edited in the CMS.
	OnBeforeUserLogin (see page 315)	This hook is called before a user login is initiated in the CMS.

Topics

Name	Description
ExtensionEvent Methods (see page 302)	The methods of the ExtensionEvent class are listed here.
ExtensionEvent Properties (see page 315)	The properties of the ExtensionEvent class are listed here.

ExtensionEvent Properties

	Name	Description
	Content (see page 315)	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 316)	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 316)	Returns object containing information about the Form, when the operation is form based.
	Site (see page 316)	Returns object containing information about the Site that fired the event.
	User (see page 316)	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.

Legend

	Method
	virtual
	Property
	protected

ExtensionEvent Methods

	Name	Description
	GetActiveDirectoryUserProxy (see page 304)	Returns an ActiveDirectoryUserProxy object to access the ActiveDirectoryUser (see page 545) Web Service.
	GetAssetProxy (see page 304)	Returns an AssetProxy object to access the Asset Web Service.
	GetBlogProxy (see page 304)	Returns a BlogProxy object to access the Blog Web Service.
	GetBusinessRulesProxy (see page 305)	Returns a BusinessRulesProxy object to access the BusinessRules (see page 54) Web Service.
	GetCalendarEventTypeProxy (see page 305)	Returns a CalendarEventTypeProxy object to access the CalendarEventType (see page 337) Web Service.
	GetCalendarProxy (see page 305)	Returns a CalendarProxy object to access the Calendar (see page 336) Web Service.
	GetContentProxy (see page 305)	Returns a ContentProxy object to access the Content (see page 315) Web Service.
	GetContentRatingProxy (see page 306)	Returns a ContentRatingProxy object to access the ContentRating (see page 379) Web Service.
	GetCustomFieldsProxy (see page 306)	Returns a CustomFieldsProxy object to access the CustomFields (see page 446) Web Service.
	GetFolderProxy (see page 306)	Returns a FolderProxy object to access the Folder Web Service.
	GetFontProxy (see page 306)	Returns a FontProxy object to access the Font Web Service.
	GetFormProxy (see page 306)	Returns a FormProxy object to access the Form Web Service.
	GetLibraryProxy (see page 307)	Returns a LibraryProxy object to access the Library (see page 475) Web Service.
	GetMetadataProxy (see page 307)	Returns a MetadataProxy object to access the Metadata (see page 495) Web Service.

	GetObjectStateAsXml (see page 307)	Retrieves changes made by individual plug-ins
	GetPermissionsProxy (see page 307)	Returns a PermissionsProxy object to access the Permissions (see page 500) Web Service.
	GetSearchManagerProxy (see page 308)	Returns a SearchManagerProxy object to access the SearchManager (see page 400) Web Service.
	GetSiteMapProxy (see page 308)	Returns a SiteMapProxy object to access the SiteMap (see page 512) Web Service.
	GetSiteProxy (see page 308)	Returns a SiteProxy object to access the Site Web Service.
	GetTaskCategoryProxy (see page 308)	Returns a TaskCategoryProxy object to access the Task Category Web Service.
	GetTaskCategoryTypeProxy (see page 308)	Returns a TaskProxy object to access the Task Web Service.
	GetTaskProxy (see page 309)	Returns a TaskProxy object to access the Task Web Service.
	GetThreadedDiscussionProxy (see page 309)	Returns a ThreadedDiscussionProxy object to access the ThreadedDiscussion (see page 390) Web Service.
	GetUserProxy (see page 309)	Returns a UserProxy object to access the User Web Service.
	OnAfterContentAdd (see page 309)	This hook is called after content (see page 337) is added in the CMS.
	OnAfterContentDelete (see page 310)	This hook is called after a delete of content (see page 337) is completed in the CMS.
	OnAfterContentEdit (see page 310)	This hook is called after an edit is completed in the CMS.
	OnAfterFolderAdd (see page 310)	This hook is called after a folder is created in the CMS.
	OnAfterFolderDelete (see page 310)	This hook is called after a folder is deleted in the CMS.
	OnAfterFolderEdit (see page 311)	This hook is called after a folder edit is completed in the CMS.
	OnAfterFormSubmit (see page 311)	This hook is called after form data is submitted to the CMS database.
	OnAfterPublish (see page 311)	This hook is called after a publish is completed in the CMS.
	OnAfterUserAdd (see page 311)	This hook is called after a user is created in the CMS.
	OnAfterUserDelete (see page 312)	This hook is called after a user is deleted from the CMS.
	OnAfterUserEdit (see page 312)	This hook is called after a user is edited in the CMS.
	OnAfterUserLogin (see page 312)	This hook is called after a user logs into in the CMS.
	OnBeforeContentAdd (see page 312)	This hook is called before content (see page 337) is added to the CMS.
	OnBeforeContentDelete (see page 312)	This hook is called before a delete of content (see page 337) in the CMS.
	OnBeforeContentEdit (see page 313)	This hook is called before a content (see page 337) is edited in the CMS.
	OnBeforeFolderAdd (see page 313)	This hook is called before a folder is created in the CMS.
	OnBeforeFolderDelete (see page 313)	This hook is called before a folder is deleted from the CMS.
	OnBeforeFolderEdit (see page 313)	This hook is called before a folder is edited in the CMS.
	OnBeforeFormSubmit (see page 314)	This hook is called before form data is submitted to the CMS database.
	OnBeforePublish (see page 314)	This hook is called before a publish takes place in the CMS.
	OnBeforeUserAdd (see page 314)	This hook is called before a user is created in the CMS.
	OnBeforeUserDelete (see page 314)	This hook is called before a user is delete from the CMS.

	OnBeforeUserEdit (see page 315)	This hook is called before a user is edited in the CMS.
	OnBeforeUserLogin (see page 315)	This hook is called before a user login is initiated in the CMS.

Legend

	Method
	virtual

ExtensionEvent Properties

	Name	Description
	Content (see page 315)	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 316)	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 316)	Returns object containing information about the Form, when the operation is form based.
	Site (see page 316)	Returns object containing information about the Site that fired the event.
	User (see page 316)	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.

Legend

	Property
	protected

ExtensionEvent Methods

The methods of the ExtensionEvent class are listed here.

Public Methods

	Name	Description
	GetActiveDirectoryUserProxy (see page 304)	Returns an ActiveDirectoryUserProxy object to access the ActiveDirectoryUser (see page 545) Web Service.
	GetAssetProxy (see page 304)	Returns an AssetProxy object to access the Asset Web Service.
	GetBlogProxy (see page 304)	Returns a BlogProxy object to access the Blog Web Service.
	GetBusinessRulesProxy (see page 305)	Returns a BusinessRulesProxy object to access the BusinessRules (see page 54) Web Service.
	GetCalendarEventTypeProxy (see page 305)	Returns a CalendarEventTypeProxy object to access the CalendarEventType (see page 337) Web Service.
	GetCalendarProxy (see page 305)	Returns a CalendarProxy object to access the Calendar (see page 336) Web Service.
	GetContentProxy (see page 305)	Returns a ContentProxy object to access the Content (see page 315) Web Service.
	GetContentRatingProxy (see page 306)	Returns a ContentRatingProxy object to access the ContentRating (see page 379) Web Service.
	GetCustomFieldsProxy (see page 306)	Returns a CustomFieldsProxy object to access the CustomFields (see page 446) Web Service.
	GetFolderProxy (see page 306)	Returns a FolderProxy object to access the Folder Web Service.
	GetFontProxy (see page 306)	Returns a FontProxy object to access the Font Web Service.
	GetFormProxy (see page 306)	Returns a FormProxy object to access the Form Web Service.
	GetLibraryProxy (see page 307)	Returns a LibraryProxy object to access the Library (see page 475) Web Service.
	GetMetadataProxy (see page 307)	Returns a MetadataProxy object to access the Metadata (see page 495) Web Service.

	GetObjectStateAsXml (see page 307)	Retrieves changes made by individual plug-ins
	GetPermissionsProxy (see page 307)	Returns a PermissionsProxy object to access the Permissions (see page 500) Web Service.
	GetSearchManagerProxy (see page 308)	Returns a SearchManagerProxy object to access the SearchManager (see page 400) Web Service.
	GetSiteMapProxy (see page 308)	Returns a SiteMapProxy object to access the SiteMap (see page 512) Web Service.
	GetSiteProxy (see page 308)	Returns a SiteProxy object to access the Site Web Service.
	GetTaskCategoryProxy (see page 308)	Returns a TaskCategoryProxy object to access the Task Category Web Service.
	GetTaskCategoryTypeProxy (see page 308)	Returns a TaskProxy object to access the Task Web Service.
	GetTaskProxy (see page 309)	Returns a TaskProxy object to access the Task Web Service.
	GetThreadedDiscussionProxy (see page 309)	Returns a ThreadedDiscussionProxy object to access the ThreadedDiscussion (see page 390) Web Service.
	GetUserProxy (see page 309)	Returns a UserProxy object to access the User Web Service.
	OnAfterContentAdd (see page 309)	This hook is called after content (see page 337) is added in the CMS.
	OnAfterContentDelete (see page 310)	This hook is called after a delete of content (see page 337) is completed in the CMS.
	OnAfterContentEdit (see page 310)	This hook is called after an edit is completed in the CMS.
	OnAfterFolderAdd (see page 310)	This hook is called after a folder is created in the CMS.
	OnAfterFolderDelete (see page 310)	This hook is called after a folder is deleted in the CMS.
	OnAfterFolderEdit (see page 311)	This hook is called after a folder edit is completed in the CMS.
	OnAfterFormSubmit (see page 311)	This hook is called after form data is submitted to the CMS database.
	OnAfterPublish (see page 311)	This hook is called after a publish is completed in the CMS.
	OnAfterUserAdd (see page 311)	This hook is called after a user is created in the CMS.
	OnAfterUserDelete (see page 312)	This hook is called after a user is deleted from the CMS.
	OnAfterUserEdit (see page 312)	This hook is called after a user is edited in the CMS.
	OnAfterUserLogin (see page 312)	This hook is called after a user logs into in the CMS.
	OnBeforeContentAdd (see page 312)	This hook is called before content (see page 337) is added to the CMS.
	OnBeforeContentDelete (see page 312)	This hook is called before a delete of content (see page 337) in the CMS.
	OnBeforeContentEdit (see page 313)	This hook is called before a content (see page 337) is edited in the CMS.
	OnBeforeFolderAdd (see page 313)	This hook is called before a folder is created in the CMS.
	OnBeforeFolderDelete (see page 313)	This hook is called before a folder is deleted from the CMS.
	OnBeforeFolderEdit (see page 313)	This hook is called before a folder is edited in the CMS.
	OnBeforeFormSubmit (see page 314)	This hook is called before form data is submitted to the CMS database.
	OnBeforePublish (see page 314)	This hook is called before a publish takes place in the CMS.
	OnBeforeUserAdd (see page 314)	This hook is called before a user is created in the CMS.
	OnBeforeUserDelete (see page 314)	This hook is called before a user is delete from the CMS.

	OnBeforeUserEdit (see page 315)	This hook is called before a user is edited in the CMS.
	OnBeforeUserLogin (see page 315)	This hook is called before a user login is initiated in the CMS.

Legend

	Method
	virtual

ExtensionEvent.GetActiveDirectoryUserProxy Method

Returns an ActiveDirectoryUserProxy object to access the ActiveDirectoryUser (see page 545) 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 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 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 54) 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 (see page 337) 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 336) Web Service.

C#

```
public CalendarProxy GetCalendarProxy();
```

Visual Basic

```
Public Function GetCalendarProxy() As CalendarProxy
```

Returns

CalendarProxy

ExtensionEvent.GetContentProxy Method

Returns a ContentProxy object to access the Content (see page 315) 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 379) 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 446) Web Service.

C#

```
public CustomFieldsProxy GetCustomFieldsProxy();
```

Visual Basic

```
Public Function GetCustomFieldsProxy() As CustomFieldsProxy
```

Returns

CustomFieldsProxy

ExtensionEvent.GetFolderProxy Method

Returns a FolderProxy object to access the Folder 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 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 Web Service.

C#

```
public FormProxy GetFormProxy();
```

Visual Basic

```
Public Function GetFormProxy() As FormProxy
```

Returns

FormProxy

ExtensionEvent.GetLibraryProxy Method

Returns a LibraryProxy object to access the Library (see page 475) Web Service.

C#

```
public LibraryProxy GetLibraryProxy();
```

Visual Basic

```
Public Function GetLibraryProxy() As LibraryProxy
```

Returns

LibraryProxy

ExtensionEvent.GetMetadataProxy Method

Returns a MetadataProxy object to access the Metadata (see page 495) 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 500) 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 400) 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 512) 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 Web Service.

C#

```
public SiteProxy GetSiteProxy();
```

Visual Basic

```
Public Function GetSiteProxy() As SiteProxy
```

Returns

SiteProxy

ExtensionEvent.GetTaskCategoryProxy Method

Returns a TaskCategoryProxy object to access the Task 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 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 Web Service.

C#

```
public TaskProxy GetTaskProxy();
```

Visual Basic

```
Public Function GetTaskProxy() As TaskProxy
```

Returns

TaskProxy

ExtensionEvent.GetThreadedDiscussionProxy Method

Returns a ThreadedDiscussionProxy object to access the ThreadedDiscussion (see page 390) 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 Web Service.

C#

```
public UserProxy GetUserProxy();
```

Visual Basic

```
Public Function GetUserProxy() As UserProxy
```

Returns

UserProxy

ExtensionEvent.OnAfterContentAdd Method

This hook is called after content (see page 337) 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 (see page 337) 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.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.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.OnBeforeContentAdd Method

This hook is called before content (see page 337) 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 (see page 337) 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 (see page 337) 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.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.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 315)	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 316)	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 316)	Returns object containing information about the Form, when the operation is form based.
	Site (see page 316)	Returns object containing information about the Site that fired the event.
	User (see page 316)	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.

Legend

	Property
	protected

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
```

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

Members

FolderInfo Properties

	Name	Description
	Description (see page 318)	Gets a string containing the description of the folder. (ReadOnly)
	ErrorMessage (see page 318)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	ID (see page 318)	Gets an integer representing the folder ID of the folder. (ReadOnly)
	Name (see page 318)	Gets a string containing the folder name. (ReadOnly)
	ParentID (see page 318)	Gets an integer representing the folder ID of the parent folder. (ReadOnly)
	ReplicationState (see page 319)	Gets an integer representing if replication is turned on for this folder. (ReadOnly)
	StyleSheet (see page 319)	Gets a string containing the style sheet to be applied to content (see page 337) in this folder. (ReadOnly)
	Type (see page 319)	Gets a string containing the folder type. (ReadOnly)

Topics

Name	Description
FolderInfo Properties (see page 317)	The properties of the FolderInfo class are listed here.

Legend

	Property

FolderInfo Properties

	Name	Description
	Description (see page 318)	Gets a string containing the description of the folder. (ReadOnly)
	ErrorMessage (see page 318)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	ID (see page 318)	Gets an integer representing the folder ID of the folder. (ReadOnly)
	Name (see page 318)	Gets a string containing the folder name. (ReadOnly)
	ParentID (see page 318)	Gets an integer representing the folder ID of the parent folder. (ReadOnly)
	ReplicationState (see page 319)	Gets an integer representing if replication is turned on for this folder. (ReadOnly)
	StyleSheet (see page 319)	Gets a string containing the style sheet to be applied to content (see page 337) in this folder. (ReadOnly)
	Type (see page 319)	Gets a string containing the folder type. (ReadOnly)

Legend

	Property

FolderInfo Properties

The properties of the FolderInfo class are listed here.

Public Properties

	Name	Description
	Description (see page 318)	Gets a string containing the description of the folder. (ReadOnly)
	ErrorMessage (see page 318)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	ID (see page 318)	Gets an integer representing the folder ID of the folder. (ReadOnly)
	Name (see page 318)	Gets a string containing the folder name. (ReadOnly)
	ParentID (see page 318)	Gets an integer representing the folder ID of the parent folder. (ReadOnly)
	ReplicationState (see page 319)	Gets an integer representing if replication is turned on for this folder. (ReadOnly)
	StyleSheet (see page 319)	Gets a string containing the style sheet to be applied to content (see page 337) in this folder. (ReadOnly)
	Type (see page 319)	Gets a string containing the folder type. (ReadOnly)

Legend

	Property

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 an integer representing the folder ID of the folder. (ReadOnly)

C#

```
public int ID;
```

Visual Basic

```
Public Property ID() As Integer
```

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 an integer representing the folder ID of the parent folder. (ReadOnly)

C#

```
public int ParentID;
```

Visual Basic

```
Public Property ParentID() As Integer
```

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 (see page 337) 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

Members

FormField Properties

	Name	Description
	FieldName (see page 320)	Gets the name of the field as a string
	FieldValue (see page 320)	Gets the value of the field as a string

Topics

Name	Description
FormField Properties (see page 320)	The properties of the FormField class are listed here.

Legend

	Property

FormField Properties

	Name	Description
	FieldName (see page 320)	Gets the name of the field as a string
	FieldValue (see page 320)	Gets the value of the field as a string

Legend

	Property

FormField Properties

The properties of the FormField class are listed here.

Public Properties

	Name	Description
	FieldName (see page 320)	Gets the name of the field as a string
	FieldValue (see page 320)	Gets the value of the field as a string

Legend

	Property

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 object

Class Hierarchy

C#

```
public sealed class FormInfo;
```

Visual Basic

```
Public sealed Class FormInfo
```

File

CmsInfoObjects.cs

Members

FormInfo Methods

	Name	Description
	this (see page 322)	Gets a single FormField (see page 319) object indexed by Field Name

Topics

Name	Description
FormInfo Methods (see page 321)	The methods of the FormInfo class are listed here.

FormInfo Properties (see page 322)	The properties of the FormInfo class are listed here.

FormInfo Properties

	Name	Description
	ErrorMessage (see page 322)	Gets or sets a string containing message to display to the end user if plug-in returns false. (ReadWrite)
	FormDescription (see page 322)	Gets the description of the form as a string
	FormFields (see page 322)	Gets a collection of the Form Field objects as a System.Collections.Generic.List
	FormTitle (see page 323)	Gets the title for the form as a string

Legend

	Indexer
	Property

FormInfo Methods

	Name	Description
	this (see page 322)	Gets a single FormField (see page 319) object indexed by Field Name

Legend

	Indexer

FormInfo Properties

	Name	Description
	ErrorMessage (see page 322)	Gets or sets a string containing message to display to the end user if plug-in returns false. (ReadWrite)
	FormDescription (see page 322)	Gets the description of the form as a string
	FormFields (see page 322)	Gets a collection of the Form Field objects as a System.Collections.Generic.List
	FormTitle (see page 323)	Gets the title for the form as a string

Legend

	Property

FormInfo Methods

The methods of the FormInfo class are listed here.

Public Methods

	Name	Description
	this (see page 322)	Gets a single FormField (see page 319) object indexed by Field Name

Legend

	Indexer

FormInfo.this Indexer

Gets a single FormField (see page 319) object indexed by Field Name

C#

```
public FormField this[string fieldName];
```

Visual Basic

```
Public Sub this(fieldName As string)
```

Parameters

Parameters	Description
`fieldName`	Name of field

FormInfo Properties

The properties of the FormInfo class are listed here.

Public Properties

	Name	Description
	ErrorMessage (see page 322)	Gets or sets a string containing message to display to the end user if plug-in returns false. (ReadWrite)
	FormDescription (see page 322)	Gets the description of the form as a string
	FormFields (see page 322)	Gets a collection of the Form Field objects as a System.Collections.Generic.List
	FormTitle (see page 323)	Gets the title for the form as a string

Legend

	Property

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 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
```

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

Members

SiteInfo Properties

	Name	Description
	AppPath (see page 324)	Gets a string containing the full application path of asp.net appication (ReadOnly)
	IsSSL (see page 324)	Gets a bool indicating if calling site uses SSL. (ReadOnly)
	Name (see page 324)	Gets a string containing the name of the calling site. (ReadOnly)
	PhysicalSitePath (see page 324)	Gets a string containing the physical path of the application (ReadOnly)
	Port (see page 325)	Gets an integer representing the port the calling site is running on. (ReadOnly)
	Url (see page 325)	Gets a string containing the URL of the calling site. (ReadOnly)

Topics

Name	Description
SiteInfo Properties (see page 324)	The properties of the SiteInfo class are listed here.

Legend

	Property

SiteInfo Properties

	Name	Description
	AppPath (see page 324)	Gets a string containing the full application path of asp.net appication (ReadOnly)
	IsSSL (see page 324)	Gets a bool indicating if calling site uses SSL. (ReadOnly)
	Name (see page 324)	Gets a string containing the name of the calling site. (ReadOnly)
	PhysicalSitePath (see page 324)	Gets a string containing the physical path of the application (ReadOnly)
	Port (see page 325)	Gets an integer representing the port the calling site is running on. (ReadOnly)
	Url (see page 325)	Gets a string containing the URL of the calling site. (ReadOnly)

Legend

	Property

SiteInfo Properties

The properties of the SiteInfo class are listed here.

Public Properties

	Name	Description
	AppPath (see page 324)	Gets a string containing the full application path of asp.net appication (ReadOnly)
	IsSSL (see page 324)	Gets a bool indicating if calling site uses SSL. (ReadOnly)
	Name (see page 324)	Gets a string containing the name of the calling site. (ReadOnly)
	PhysicalSitePath (see page 324)	Gets a string containing the physical path of the application (ReadOnly)
	Port (see page 325)	Gets an integer representing the port the calling site is running on. (ReadOnly)
	Url (see page 325)	Gets a string containing the URL of the calling site. (ReadOnly)

Legend

	Property

SiteInfo.AppPath Property

Gets a string containing the full application path of asp.net appication (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

C#

```
public sealed class UserInfo;
```

Visual Basic

```
Public sealed Class UserInfo
```

File

CmsInfoObjects.cs

Members

UserInfo Properties

	Name	Description
	DisplayName (see page 326)	Gets a string containing the display name of the user being added, edited or deleted. (ReadOnly)
	Editor (see page 327)	Gets a string containing the editor preference of the user being added, edited or deleted. (ReadOnly)
	EmailAddress (see page 327)	Gets a string containing the email address of the user being added, edited or deleted. (ReadOnly)
	ErrorMessage (see page 327)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	FirstName (see page 327)	Gets a string containing the first name of the user being added, edited or deleted. (ReadOnly)
	ID (see page 327)	Gets an integer representing the user ID of the being added, edited or deleted. (ReadOnly)
	Language (see page 327)	Gets a string containing the language preference of the user being added, edited or deleted. (ReadOnly)
	LastName (see page 328)	Gets a string containing the last name of the user being added, edited or deleted. (ReadOnly)
	Notifications (see page 328)	Gets an integer representing if the user has turned on email notifications of the user being added, edited or deleted. (ReadOnly)
	Type (see page 328)	Integer representing the user type of the user being added, edited or deleted. Only valid with OnBeforeUserAdd and OnAfterUserAdd. (ReadOnly)
	UserName (see page 328)	Gets a string containing the login name of the user being added, edited or deleted. (ReadOnly)

Topics

Name	Description
UserInfo Properties (see page 326)	The properties of the UserInfo class are listed here.

Legend

	Property

UserInfo Properties

	Name	Description
	DisplayName (see page 326)	Gets a string containing the display name of the user being added, edited or deleted. (ReadOnly)
	Editor (see page 327)	Gets a string containing the editor preference of the user being added, edited or deleted. (ReadOnly)
	EmailAddress (see page 327)	Gets a string containing the email address of the user being added, edited or deleted. (ReadOnly)
	ErrorMessage (see page 327)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	FirstName (see page 327)	Gets a string containing the first name of the user being added, edited or deleted. (ReadOnly)
	ID (see page 327)	Gets an integer representing the user ID of the being added, edited or deleted. (ReadOnly)
	Language (see page 327)	Gets a string containing the language preference of the user being added, edited or deleted. (ReadOnly)
	LastName (see page 328)	Gets a string containing the last name of the user being added, edited or deleted. (ReadOnly)
	Notifications (see page 328)	Gets an integer representing if the user has turned on email notifications of the user being added, edited or deleted. (ReadOnly)
	Type (see page 328)	Integer representing the user type of the user being added, edited or deleted. Only valid with OnBeforeUserAdd and OnAfterUserAdd. (ReadOnly)
	UserName (see page 328)	Gets a string containing the login name of the user being added, edited or deleted. (ReadOnly)

Legend

	Property

UserInfo Properties

The properties of the UserInfo class are listed here.

Public Properties

	Name	Description
	DisplayName (see page 326)	Gets a string containing the display name of the user being added, edited or deleted. (ReadOnly)
	Editor (see page 327)	Gets a string containing the editor preference of the user being added, edited or deleted. (ReadOnly)
	EmailAddress (see page 327)	Gets a string containing the email address of the user being added, edited or deleted. (ReadOnly)
	ErrorMessage (see page 327)	Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)
	FirstName (see page 327)	Gets a string containing the first name of the user being added, edited or deleted. (ReadOnly)
	ID (see page 327)	Gets an integer representing the user ID of the being added, edited or deleted. (ReadOnly)
	Language (see page 327)	Gets a string containing the language preference of the user being added, edited or deleted. (ReadOnly)
	LastName (see page 328)	Gets a string containing the last name of the user being added, edited or deleted. (ReadOnly)
	Notifications (see page 328)	Gets an integer representing if the user has turned on email notifications of the user being added, edited or deleted. (ReadOnly)
	Type (see page 328)	Integer representing the user type of the user being added, edited or deleted. Only valid with OnBeforeUserAdd and OnAfterUserAdd. (ReadOnly)
	UserName (see page 328)	Gets a string containing the login name of the user being added, edited or deleted. (ReadOnly)

Legend

	Property

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 an integer representing the user ID of the being added, edited or deleted. (ReadOnly)

C#

```
public int ID;
```

Visual Basic

```
Public Property ID() As Integer
```

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.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#

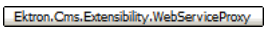
```
public string UserName;
```

Visual Basic

```
Public Property UserName() As string
```

WebServiceProxy Class

Class Hierarchy



C#

```
internal class WebServiceProxy;
```

Visual Basic

```
Friend Class WebServiceProxy
```

File

ExtensibilityBase.cs

Description

This is class Ektron.Cms.Extensibility.WebServiceProxy.

.NET Assemblies

The .NET Assemblies documentation details the methods and properties that are available for customizing a site in CMS400.NET.

Namespaces

Name	Description
API (see page 329)	This is namespace API.

API Namespace

This is namespace API.

Namespaces

Name	Description
Analytics (see page 330)	This is namespace API.Analytics.
Calendar (see page 336)	This is namespace API.Calendar.
Content (see page 337)	This is namespace API.Content.
Search (see page 399)	This is namespace API.Search.
User (see page 402)	This is namespace API.User.

Classes

	Name	Description
	BusinessRule (see page 444)	This is class API.BusinessRule.
	CustomFields (see page 446)	This is class API.CustomFields.
	DateTimeSelector (see page 458)	This is class API.DateTimeSelector.
	Folder (see page 466)	This is class API.Folder.
	Font (see page 472)	This is class API.Font.
	Library (see page 475)	This is class API.Library.
	Localization (see page 484)	This is class API.Localization.
	Metadata (see page 495)	This is class API.Metadata.
	Permissions (see page 500)	This is class API.Permissions.
	Site (see page 505)	This is class API.Site.
	SiteMap (see page 512)	This is class API.SiteMap.
	Subscription (see page 515)	This is class API.Subscription.
	Utilities (see page 524)	This is class API.Utilities.

Legend

	Class

API.Analytics Namespace

This is namespace API.Analytics.

Classes

	Name	Description
	Analytics (see page 330)	A class for managing and querying CMS400.NET site activity information

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	Analytics (see page 330)	A class for managing and querying CMS400.NET site activity information

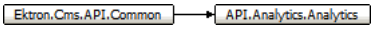
Legend

	Class

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

Members

Analytics Methods

	Name	Description
	GetSessionRank (see page 332)	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 332)	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 332)	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 database and not to the standard CMS database.
	SoftDumpAnalyticsBuffer (see page 333)	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 333)	Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer. This method is overloaded.

Topics

Name	Description
Analytics Methods (see page 332)	The methods of the Analytics class are listed here.
Analytics Properties (see page 334)	The properties of the Analytics class are listed here.

Analytics Properties

	Name	Description
	AnalyticsEnabled (see page 334)	Indicates if Analytics 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 12) server control.
	AppendToIISLog (see page 335)	Indicates whether or not the web.config has been set to append user_id and content_id information to the IIS log.
	InMemoryContentHitsData (see page 335)	The dataset that stores all the page/content (see page 337) 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.
	RecordsBeforeWrite (see page 335)	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 335)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.
	TimeBeforeWrite (see page 335)	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 335)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.

Legend

	Method
	Property
	read only

Analytics Methods

	Name	Description
	GetSessionRank (see page 332)	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 332)	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 332)	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 database and not to the standard CMS database.
	SoftDumpAnalyticsBuffer (see page 333)	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 333)	Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer. This method is overloaded.

Legend

	Method

Analytics Properties

	Name	Description
	AnalyticsEnabled (see page 334)	Indicates if Analytics 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 12) server control.
	AppendToIISLog (see page 335)	Indicates whether or not the web.config has been set to append user_id and content_id information to the IIS log.
	InMemoryContentHitsData (see page 335)	The dataset that stores all the page/content (see page 337) 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.
	RecordsBeforeWrite (see page 335)	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 335)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.
	TimeBeforeWrite (see page 335)	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 335)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.

Legend

	Property
	read only

Analytics Methods

The methods of the Analytics class are listed here.

Public Methods

	Name	Description
	GetSessionRank (see page 332)	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 332)	Forces the analytics (see page 330) 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 332)	Provides a simple way of querying the database for analytics (see page 330) data. Returns the result as a weakly typed DataSet. Note that the connection is to the Analytics (see page 330) database and not to the standard CMS database.
	SoftDumpAnalyticsBuffer (see page 333)	Conditionally writes the analytics (see page 330) 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 333)	Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer. This method is overloaded.

Legend

	Method

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
page	

Analytics.HardDumpAnalyticsBuffer Method

Forces the analytics (see page 330) 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 (see page 330) data. Returns the result as a weakly typed DataSet. Note that the connection is to the Analytics (see page 330) 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
selectCommands	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 (see page 330) 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

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 (Integer, Integer, String, Integer, String, String) (see page 333)	Tracks a visit to a page, and places it into the analytics (see page 330) buffer. This method calls SoftDumpAnalyticsBuffer (see page 333).
	Analytics.TrackVisit (System.Web.UI.Page, Integer) (see page 334)	Tracks a visit to a page, and places it into the analytics (see page 330) buffer. This method calls SoftDumpAnalyticsBuffer (see page 333).

Analytics.TrackVisit Method (Integer, Integer, String, Integer, String, String)

Tracks a visit to a page, and places it into the analytics (see page 330) buffer. This method calls SoftDumpAnalyticsBuffer (see page 333).

C#

```
public TrackVisit(int content_id, int user_id, String visitor_id, int session_rank, String
referring_url, String url);
```

Visual Basic

```
Public Sub TrackVisit(ByVal content_id As Integer, ByVal user_id As Integer, ByVal visitor_id
As String, ByVal session_rank As Integer, ByVal referring_url As String, ByVal url As String)
```

Parameters

Parameters	Description
content_id	The ID of a content (see page 337) block that a visitor has viewed.
user_id	The CMS user ID of either a CMS User or Member. Is zero for a visitor.
visitor_id	A unique ID used for tracking purposes.
session_rank	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.

referring_url	An external URL a user has arrived from.
url	The URL the user is visiting.

Analytics.TrackVisit Method (System.Web.UI.Page, Integer)

Tracks a visit to a page, and places it into the analytics (see page 330) buffer. This method calls SoftDumpAnalyticsBuffer (see page 333).

C#

```
public TrackVisit(System.Web.UI.Page page, int content_id);
```

Visual Basic

```
Public Sub TrackVisit(ByVal page As System.Web.UI.Page, ByVal content_id As Integer)
```

Parameters

Parameters	Description
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.
content_id	The ID of a content (see page 337) block that a visitor has viewed.

Analytics Properties

The properties of the Analytics class are listed here.

Public Properties

	Name	Description
	AnalyticsEnabled (see page 334)	Indicates if Analytics (see page 330) has been enabled through a setting in the web.config file. If disabled, no analytics (see page 330) 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 12) server control.
	AppendToIISLog (see page 335)	Indicates whether or not the web.config has been set to append user_id and content_id information to the IIS log.
	InMemoryContentHitsData (see page 335)	The dataset that stores all the page/content (see page 337) 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.
	RecordsBeforeWrite (see page 335)	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 335)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.
	TimeBeforeWrite (see page 335)	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 335)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.

Legend

	Property
	read only

Analytics.AnalyticsEnabled Property

Indicates if Analytics (see page 330) has been enabled through a setting in the web.config file. If disabled, no analytics (see page 330) 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 12) server control.

C#

```
public ReadOnly Boolean AnalyticsEnabled;
```

Visual Basic

```
Public ReadOnly Property AnalyticsEnabled() As Boolean
```

Description

True indicates that analytics (see page 330) data is being tracked. False indicates that no analytics (see page 330) 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 (see page 337) 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 335)' 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 335)' 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
```

Copyright (c) Ektron Inc. 2006

API.Calendar Namespace

This is namespace API.Calendar.

Classes

	Name	Description
	Calendar (see page 336)	This is class API.Calendar.Calendar.
	CalendarEvent (see page 337)	This is class API.Calendar.CalendarEvent.
	CalendarEventType (see page 337)	This is class API.Calendar.CalendarEventType.

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

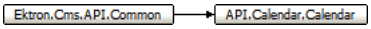
	Name	Description
	Calendar (see page 336)	This is class API.Calendar.Calendar.
	CalendarEvent (see page 337)	This is class API.Calendar.CalendarEvent.
	CalendarEventType (see page 337)	This is class API.Calendar.CalendarEventType.

Legend

	Class

Calendar Class

Class Hierarchy



C#

```
public class Calendar : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Calendar
Inherits Ektron.Cms.API.Common
```

File

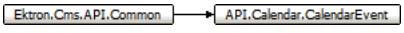
Calendar.vb

Description

This is class API.Calendar.Calendar.

CalendarEvent Class

Class Hierarchy



C#

```
public class CalendarEvent : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class CalendarEvent
Inherits Ektron.Cms.API.Common
```

File

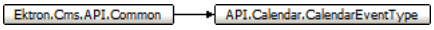
CalendarEvent.vb

Description

This is class API.Calendar.CalendarEvent.

CalendarEventType Class

Class Hierarchy



C#

```
public class CalendarEventType : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class CalendarEventType
Inherits Ektron.Cms.API.Common
```

File

CalendarEventType.vb

Description

This is class API.Calendar.CalendarEventType.

API.Content Namespace

This is namespace API.Content.

Classes

	Name	Description
	Asset (see page 338)	This is class API.Content.Asset.
	Blog (see page 344)	This is class API.Content.Blog.
	Content (see page 354)	This is class API.Content.Content.
	ContentRating (see page 379)	This is class API.Content.ContentRating.
	Form (see page 383)	This is class API.Content.Form.
	ThreadedDiscussion (see page 390)	This is class API.Content.ThreadedDiscussion.

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	Asset (see page 338)	This is class API.Content.Asset.
	Blog (see page 344)	This is class API.Content.Blog.
	Content (see page 354)	This is class API.Content.Content.
	ContentRating (see page 379)	This is class API.Content.ContentRating.
	Form (see page 383)	This is class API.Content.Form.
	ThreadedDiscussion (see page 390)	This is class API.Content.ThreadedDiscussion.

Legend

	Class

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 API.Content.Asset.

Members

Asset Methods

	Name	Description
	GetAssetMgtConfigInfo (see page 340)	Loads all of the registered plug-in configuration details.
	GetAssetMgtInfo (see page 340)	Loads all of the registered plug-in type details.
	GetAssetSupertypes (see page 340)	Loads all of the asset supertypes supported by documents add-in.
	GetCreateDragDropSnippet (see page 341)	Loads the drag and drop client snippet used with the Document Management functionality.
	GetCreateEktronExplorerSnippet (see page 341)	Loads the explorer snippet used with the Document Management functionality.
	GetCreateSnippet (see page 341)	Loads the create document snippet used with the Document Management functionality.
	GetDragDropWOSnippet (see page 341)	Loads the edit snippet for the given document when working offline. Used with the Document Management functionality.
	GetEditSnippet (see page 342)	Loads the edit snippet for the given document. Used with the Document Management functionality.
	GetMyAssetMgtID (see page 342)	Retrieves the Document Management functionality client ID.
	GetViewUrl (see page 342)	Returns Url associated with the current asset
	MakeCopyDMSAsset (see page 343)	Create a copy of DMS Asset.

	SetAssetMgtConfigInfo (see page 343)	Updates the Document Management configuration information.

Topics

Name	Description
Asset Methods (see page 339)	The methods of the Asset class are listed here.

Legend

	Method

Asset Methods

	Name	Description
	GetAssetMgtConfigInfo (see page 340)	Loads all of the registered plug-in configuration details.
	GetAssetMgtInfo (see page 340)	Loads all of the registered plug-in type details.
	GetAssetSupertypes (see page 340)	Loads all of the asset supertypes supported by documents add-in.
	GetCreateDragDropSnippet (see page 341)	Loads the drag and drop client snippet used with the Document Management functionality.
	GetCreateEktronExplorerSnippet (see page 341)	Loads the explorer snippet used with the Document Management functionality.
	GetCreateSnippet (see page 341)	Loads the create document snippet used with the Document Management functionality.
	GetDragDropWOSnippet (see page 341)	Loads the edit snippet for the given document when working offline. Used with the Document Management functionality.
	GetEditSnippet (see page 342)	Loads the edit snippet for the given document. Used with the Document Management functionality.
	GetMyAssetMgtID (see page 342)	Retrieves the Document Management functionality client ID.
	GetViewUrl (see page 342)	Returns Url associated with the current asset
	MakeCopyDMSAsset (see page 343)	Create a copy of DMS Asset.
	SetAssetMgtConfigInfo (see page 343)	Updates the Document Management configuration information.

Legend

	Method

Asset Methods

The methods of the Asset class are listed here.

Public Methods

	Name	Description
	GetAssetMgtConfigInfo (see page 340)	Loads all of the registered plug-in configuration details.
	GetAssetMgtInfo (see page 340)	Loads all of the registered plug-in type details.
	GetAssetSupertypes (see page 340)	Loads all of the asset supertypes supported by documents add-in.
	GetCreateDragDropSnippet (see page 341)	Loads the drag and drop client snippet used with the Document Management functionality.
	GetCreateEktronExplorerSnippet (see page 341)	Loads the explorer snippet used with the Document Management functionality.
	GetCreateSnippet (see page 341)	Loads the create document snippet used with the Document Management functionality.
	GetDragDropWOSnippet (see page 341)	Loads the edit snippet for the given document when working offline. Used with the Document Management functionality.
	GetEditSnippet (see page 342)	Loads the edit snippet for the given document. Used with the Document Management functionality.

	GetMyAssetMgtID (see page 342)	Retrieves the Document Management functionality client ID.
	GetViewUrl (see page 342)	Returns Url associated with the current asset
	MakeCopyDMSAsset (see page 343)	Create a copy of DMS Asset.
	SetAssetMgtConfigInfo (see page 343)	Updates the Document Management configuration information.

Legend

	Method

Asset.GetAssetMgtConfigInfo Method

Loads all of the registered plug-in configuration details.

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.

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 documents add-in.

C#

```
public AssetInfoData GetAssetSupertypes();
```

Visual Basic

```
Public Function GetAssetSupertypes() As AssetInfoData
```

Returns

Ektron.Cms.AssetInfoData()

Asset.GetCreateDragDropSnippet Method

Loads the drag and drop client snippet used with the Document Management functionality.

C#

```
public String GetCreateDragDropSnippet();
```

Visual Basic

```
Public Function GetCreateDragDropSnippet() As String
```

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetCreateEktronExplorerSnippet Method

Loads the explorer snippet used with the Document Management functionality.

C#

```
public String GetCreateEktronExplorerSnippet();
```

Visual Basic

```
Public Function GetCreateEktronExplorerSnippet() As String
```

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetCreateSnippet Method

Loads the create document snippet used with the Document Management functionality.

C#

```
public String GetCreateSnippet(int ContentType);
```

Visual Basic

```
Public Function GetCreateSnippet(ByVal ContentType As Integer) As String
```

Parameters

Parameters	Description
ContentType	The type of the content (see page 337), 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.GetDragDropWOSnippet Method

Loads the edit snippet for the given document when working offline. Used with the Document Management functionality.

C#

```
public String GetDragDropWOSnippet(String AssetId, int ContentType);
```

Visual Basic

```
Public Function GetDragDropWOSnippet(ByVal AssetId As String, ByVal ContentType As Integer) As
String
```

Parameters

Parameters	Description
AssetId	The document's ID (Required)
ContentType	The type of the content (see page 337), 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.GetEditSnippet Method

Loads the edit snippet for the given document. Used with the Document Management functionality.

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
AssetId	The document's ID (Required)
ContentType	The type of the content (see page 337), 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.

C#

```
public String GetMyAssetMgtID();
```

Visual Basic

```
Public Function GetMyAssetMgtID() As String
```

Returns

String

Asset.GetViewUrl Method

Returns Url associated with the current asset

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
AssetId	The document's ID (Required)
ContentType	The type of the content (see page 337), 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 DMS Asset.

C#

```
public Boolean MakeCopyDMSAsset(String strAssetID, String strAssetVersion);
```

Visual Basic

```
Public Function MakeCopyDMSAsset(ByRef strAssetID As String, ByRef strAssetVersion As String)
As Boolean
```

Parameters

Parameters	Description
strAssetID	ID associated with the asset
strAssetVersion	Version of the asset

Returns

boolean

Remarks

User login with create content (see page 337) privileges are required to perform this operation.

Asset.SetAssetMgtConfigInfo Method

Updates the Document Management configuration information.

C#

```
public Boolean SetAssetMgtConfigInfo(AssetConfigInfo asset_config, ));
```

Visual Basic

```
Public Function SetAssetMgtConfigInfo(ByVal asset_config As AssetConfigInfo,  As )) As Boolean
```

Parameters

Parameters	Description
asset_config	The Document Management configuration data

Returns

Boolean

Remarks

A user with administrator privileges must be logged in to perform this operation.

Blog Class

Class Hierarchy

C#

```
public class Blog : Content;
```

Visual Basic

```
Public Class Blog
Inherits Content
```

File

Blog.vb

Description

This is class API.Content.Blog.

Members

Blog Methods

	Name	Description
	AddBlog (see page 346)	Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.
	AddCommentForPost (see page 346)	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 346)	Adds a post to a given blog. Accepts a ContentData class as the post content (see page 337). Return new post ID.
	DeleteBlog (see page 347)	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.
	DeletePost (see page 347)	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 347)	Returns a blank BlogPostData class. Used Primarily in the Workarea.
	GetBlog (see page 348)	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.
	GetBlogData (see page 349)	Overloaded Method to obtain a BlogData Class.
	GetBlogDataForPost (see page 352)	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 352)	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 352)	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.
	GetPost (see page 354)	Gets a blog post for a given ID. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData Class.

Topics

Name	Description
Blog Methods (see page 345)	The methods of the Blog class are listed here.

Legend

	Method

Blog Methods

	Name	Description
	AddBlog (see page 346)	Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.

	AddCommentForPost (see page 346)	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 346)	Adds a post to a given blog. Accepts a ContentData class as the post content (see page 337). Return new post ID.
	DeleteBlog (see page 347)	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.
	DeletePost (see page 347)	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 347)	Returns a blank BlogPostData class. Used Primarily in the Workarea.
	GetBlog (see page 348)	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.
	GetBlogData (see page 349)	Overloaded Method to obtain a BlogData Class.
	GetBlogDataForPost (see page 352)	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 352)	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 352)	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.
	GetPost (see page 354)	Gets a blog post for a given ID. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData Class.

Legend

	Method

Blog Methods

The methods of the Blog class are listed here.

Public Methods

	Name	Description
	AddBlog (see page 346)	Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.
	AddCommentForPost (see page 346)	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 346)	Adds a post to a given blog. Accepts a ContentData class as the post content (see page 337). Return new post ID.
	DeleteBlog (see page 347)	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.
	DeletePost (see page 347)	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 347)	Returns a blank BlogPostData class. Used Primarily in the Workarea.
	GetBlog (see page 348)	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.
	GetBlogData (see page 349)	Overloaded Method to obtain a BlogData Class.
	GetBlogDataForPost (see page 352)	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 352)	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 352)	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.
	GetPost (see page 354)	Gets a blog post for a given ID. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData Class.

Legend

	Method

Blog.AddBlog Method

Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.

C#

```
public int AddBlog(int 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 Integer, 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,  As ), ByVal RollLinks As BlogRoll) As Integer
```

Parameters

Parameters	Description
ParentFolderID	
BlogName	Name of the Blog.
BlogTitle	Title of the Blog.
BlogDescription	Description of the Blog.
Visibility	Is the blog public/private?
EnableComments	Allow commenting.
ModerateComments	If commenting is allowed, moderate comments.
RequireAuthentication	If commenting is allowed, require authentication to add comment.
Categories	String Array representing the categories for this blog.
RollLinks	The Blog's roll links.

Returns

Integer 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
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 (see page 337). Return new post ID.

C#

```
public int AddPost(int blogID, ContentData PostContent, String Categories, ), Boolean
```

```
Pingback, String Tags, String TrackbackURL);
```

Visual Basic

```
Public Function AddPost(ByVal blogID As Integer, ByVal PostContent As ContentData, ByVal
Categories As String,  As ), ByVal Pingback As Boolean, ByVal Tags As String, ByVal
TrackbackURL As String) As Integer
```

Parameters

Parameters	Description
blogID	Integer ID of the blog to add the post to.
PostContent	ContentData class representing the HTML, title, etc.
Pingback	Pingback URL(s) in this post?
Tags	String list of tags that that applies to this post.
TrackbackURL	Trackback URL(s) to notify.
CatgetBlogApiInstanceegories	String Array representing the categories this post applies to.

Returns

Integer 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.

C#

```
public DeleteBlog(int BlogID);
```

Visual Basic

```
Public Sub DeleteBlog(ByVal BlogID As Integer)
```

Parameters

Parameters	Description
BlogID	The integer ID of the blog. This corresponds to the folder id of the blog.

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(int post_id);
```

Visual Basic

```
Public Function DeletePost(ByVal post_id As Integer) As Boolean
```

Parameters

Parameters	Description
post_id	Integer value of the post ID you wish 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

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.

Overload List

	Name	Description
	Blog.GetBlog (Integer, String, Integer, Integer) (see page 348)	Returns a BlogData class for a given blog id, with parameters. This class contains the content (see page 337) 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 349)	Returns a BlogData class for a given path, with parameters. This class contains the content (see page 337) for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

Blog.GetBlog Method (Integer, String, Integer, Integer)

Returns a BlogData class for a given blog id, with parameters. This class contains the content (see page 337) for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

C#

```
public BlogData GetBlog(int ID, String StartDate, int MaxEntries, int LanguageID);
```

Visual Basic

```
Public Function GetBlog(ByVal ID As Integer, ByVal StartDate As String, ByVal MaxEntries As
Integer, ByVal LanguageID As Integer) As BlogData
```

Parameters

Parameters	Description
ID	The id of the blog. This is a integer value that corresponds to the folder id of the blog.
StartDate	The date to get entries from. Used if you want to get historical information. Gets information only for the date specified.
MaxEntries	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 (see page 337) is not returned, only the post info for the content (see page 337).
LanguageID	The language ID for the content (see page 337) 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.

Blog.GetBlog Method (String, String, Integer, Integer)

Returns a BlogData class for a given path, with parameters. This class contains the content (see page 337) 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
`path`	The path of the blog. This is a string value that corresponds to the folder path of the blog.
`StartDate`	The date to get entries from. Used if you want to get historical information. Gets information only for the date specified.
`MaxEntries`	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 (see page 337) is not returned, only the post info for the content (see page 337).
`LanguageID`	The language ID for the content (see page 337) to get. 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

Overloaded Method to obtain a BlogData Class.

Overload List

	Name	Description
	Blog.GetBlogData (FolderData) (see page 350)	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: <link ContentAPI.BlogObject@Integer@DateTime, BlogObject(Integer DateTime)> <link ContentAPI.BlogObject@Integer@DateTime@Integer@Integer, BlogObject(Integer DateTime Integer Integer)> <link ContentAPI.BlogObject@Integer, BlogObject(Integer)>
	Blog.GetBlogData (Integer) (see page 350)	Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog. Returns either a populated BlogData class or null (nothing). This function is overloaded. See Also: <link ContentAPI.BlogObject@Integer@DateTime, BlogObject(Integer DateTime)> <link ContentAPI.BlogObject@Integer@DateTime@Integer@Integer, BlogObject(Integer DateTime Integer Integer)> <link ContentAPI.BlogObject@FolderData, BlogObject(FolderData)>

	Blog.GetBlogData (Integer, DateTime) (see page 351)	Overloaded Method to obtain a BlogData Class. Accepts the integer 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: <link ContentAPI.BlogObject@Integer, BlogObject(Integer)> <link ContentAPI.BlogObject@Integer@DateTime@Integer@Integer, BlogObject(Integer DateTime Integer Integer)> <link ContentAPI.BlogObject@FolderData, BlogObject(FolderData)>
	Blog.GetBlogData (Integer, DateTime, Integer, Integer) (see page 351)	Overloaded Method to obtain a BlogData Class. Accepts the integer 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: <link ContentAPI.BlogObject@Integer, BlogObject(Integer)> <link ContentAPI.BlogObject@Integer@DateTime, BlogObject(Integer DateTime)> <link ContentAPI.BlogObject@FolderData, 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:

<link ContentAPI.BlogObject@Integer@DateTime, BlogObject(Integer DateTime)>

<link ContentAPI.BlogObject@Integer@DateTime@Integer@Integer, BlogObject(Integer DateTime Integer Integer)>

<link ContentAPI.BlogObject@Integer, BlogObject(Integer)>

C#

```
public BlogData GetBlogData(FolderData folder_data);
```

Visual Basic

```
Public Function GetBlogData(ByVal folder_data As FolderData) As BlogData
```

Parameters

Parameters	Description
folder_data	A populated FolderData class. (see FolderData)

Returns

A populated BlogData class or nothing

Blog.GetBlogData Method (Integer)

Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

<link ContentAPI.BlogObject@Integer@DateTime, BlogObject(Integer DateTime)>

<link ContentAPI.BlogObject@Integer@DateTime@Integer@Integer, BlogObject(Integer DateTime Integer Integer)>

<link ContentAPI.BlogObject@FolderData, BlogObject(FolderData)>

C#

```
public BlogData GetBlogData(int BlogID);
```

Visual Basic

```
Public Function GetBlogData(ByVal BlogID As Integer) As BlogData
```

Parameters

Parameters	Description
BlogID	Integer ID of the blog. This corresponds to the id of the blog folder.

Returns

Either a populated BlogData class or null (nothing).

Blog.GetBlogData Method (Integer, DateTime)

Overloaded Method to obtain a BlogData Class. Accepts the integer 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:

<link ContentAPI.BlogObject@Integer, BlogObject(Integer)>

<link ContentAPI.BlogObject@Integer@DateTime@Integer@Integer, BlogObject(Integer DateTime Integer Integer)>

<link ContentAPI.BlogObject@FolderData, BlogObject(FolderData)>

C#

```
public BlogData GetBlogData(int BlogID, DateTime StartDate);
```

Visual Basic

```
Public Function GetBlogData(ByVal BlogID As Integer, ByVal StartDate As DateTime) As BlogData
```

Parameters

Parameters	Description
BlogID	Integer ID of the blog. This corresponds to the id of the blog folder.
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).

Blog.GetBlogData Method (Integer, DateTime, Integer, Integer)

Overloaded Method to obtain a BlogData Class. Accepts the integer 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:

<link ContentAPI.BlogObject@Integer, BlogObject(Integer)>

<link ContentAPI.BlogObject@Integer@DateTime, BlogObject(Integer DateTime)>

<link ContentAPI.BlogObject@FolderData, BlogObject(FolderData)>

C#

```
public BlogData GetBlogData(int BlogID, DateTime StartDate, int Postsvisible, int LanguageID);
```

Visual Basic

```
Public Function GetBlogData(ByVal BlogID As Integer, ByVal StartDate As DateTime, ByVal
```

```
Postsvisible As Integer, ByVal LanguageID As Integer) As BlogData
```

Parameters

Parameters	Description
BlogID	Integer ID of the blog. This corresponds to the id of the blog folder.
StartDate	The DateTime value to get posts from. You can obtain historical posts with this property, or leave it blank to begin now.
Postsvisible	Integer number of posts to return. Leave as 0 to use the blog setting, or -1 to return none.
LanguageID	Integer ID of the language.

Returns

Either a populated BlogData class or null (nothing).

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(int post_id);
```

Visual Basic

```
Public Function GetBlogDataForPost(ByVal post_id As Integer) As BlogData
```

Parameters

Parameters	Description
post_id	Integer 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(int id);
```

Visual Basic

```
Public Function GetBlogPostData(ByVal id As Integer) As BlogPostData
```

Parameters

Parameters	Description
id	Integer value of the post ID

Returns

BlogPostData Class.

GetBlogRoll Method

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 (Integer) (see page 353)	Returns a populated blog roll class. Accepts a blog ID integer value. See the BlogRoll (see page 40) class for more information.
	Blog.GetBlogRoll (String) (see page 353)	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 40) class for the properties and usage of the blog roll. This is an overloaded method. See Also: LoadBlogRoll (Integer) Method.

Blog.GetBlogRoll Method (Integer)

Returns a populated blog roll class. Accepts a blog ID integer value. See the BlogRoll (see page 40) class for more information.

C#

```
public BlogRoll GetBlogRoll(int blog_id);
```

Visual Basic

```
Public Function GetBlogRoll(ByVal blog_id As Integer) As BlogRoll
```

Parameters

Parameters	Description
`blog_id`	Integer ID of the blog. This corresponds to the folder ID of the blog.

Returns

A populated BlogRoll (see page 40) 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 40) class for the properties and usage of the blog roll.

This is an overloaded method. See Also: LoadBlogRoll (Integer) Method.

C#

```
public BlogRoll GetBlogRoll(String Path);
```

Visual Basic

```
Public Function GetBlogRoll(ByVal Path As String) As BlogRoll
```

Parameters

Parameters	Description
`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 40) 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(int post_id, BlogPostData PostInfo);
```

Visual Basic

```
Public Function GetPost(ByVal post_id As Integer, ByRef PostInfo As BlogPostData) As
ContentData
```

Parameters

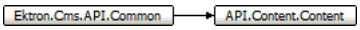
Parameters	Description
post_id	Integer ID for the post.
PostInfo	BlogPostData class representing the blog specific post info. This is also in the metadata.

Returns

ContentData class with a ByRef BlogPostData

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 API.Content.Content.

Members

Content Methods

	Name	Description
	AddContent (see page 360)	Adds Content to the CMS. This method is overloaded.
	AddTaxonomyItem (see page 362)	
	AddTaxonomySyncFolder (see page 363)	adds folder entry into taxonomy
	AddXmlConfiguration (see page 363)	Adds an XML configuration for XML oriented content.
	CheckOutContent (see page 363)	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 364)	Copy the content into required folder.
	CreateTaxonomy (see page 364)	creates new taxonomy
	DeleteContentItem (see page 364)	Delete the content by the content's ID.

	DeleteTaxonomy (see page 365)	deletes the existing taxonomy
	DeleteXmlConfiguration (see page 365)	Deletes an XML configuration.
	GetAddViewLanguage (see page 365)	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 365)	Loads all of the comments.
	GetAllTemplates (see page 366)	Loads all of the templates.
	GetAllUnassignedItemApprovals (see page 366)	Returns all of the unassigned user/groups for a given content ID.
	GetAllXmlConfigurations (see page 367)	Reads all of the XML configurations.
	GetChildContent (see page 367)	Loads all of the contents for the given folder.
	GetContent (see page 367)	Returns list of content. This function is overloaded.
	GetContentByHistory (see page 369)	Retrieves the corresponding content by the history ID.
	GetContentDisplayLink (see page 369)	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 370)	Loads the content details with checkout mode.
	GetContentReport (see page 370)	Loads the content data for the given constraints.
	GetContentState (see page 370)	Loads the content's details.
	GetContentStatus (see page 371)	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 371)	Gets web domain for a piece of content.
	GetExpireContent (see page 371)	Loads all of the expired content.
	GetTaxonomyIdByPath (see page 372)	returns taxonomy id by given path
	LoadTaxonomy (see page 372)	returns the TaxonomyData with all its subcategory and its items.
	MoveContentToFolder (see page 372)	Moves content to a folder.
	PublishContent (see page 373)	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.
	ReadAllAssignedCategory (see page 373)	returns all the assigned category for the content id
	ReadCategoryByTaxonomy (see page 374)	reads all the subcategory for given taxonomy
	ReadTaxonomy (see page 374)	read the taxonomy data
	RelicateXmlConfiguration (see page 374)	Relicate the XML configuration.
	RemoveTaxonomyFolder (see page 375)	removes the sync up folder from taxonomy
	RemoveTaxonomyItem (see page 375)	removes item from the taxonomy
	RenameContent (see page 375)	Renames the content.
	ReOrderTaxonomyItems (see page 375)	reorder display of taxonomy items
	RestoreHistoryContent (see page 376)	Restores the content to a particular version of content.

	SaveContent (see page 376)	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 377)	Delete the content by the content's Id and folder Id.
	UndoCheckout (see page 377)	This undoes the checkout for a piece of content which is checked out to the current user.
	UpdateContentMetaData (see page 377)	Updates a single piece of content metadata
	UpdateTaxonomy (see page 378)	updates the existing taxonomy
	UpdatexmlCollectionPackage (see page 378)	Updates the XML collection package.
	UpdateXmlConfiguration (see page 378)	Updates the XML configuration details.

Topics

Name	Description
Content Methods (see page 358)	The methods of the Content class are listed here.

Legend

	Method

Content Methods

	Name	Description
	AddContent (see page 360)	Adds Content to the CMS. This method is overloaded.
	AddTaxonomyItem (see page 362)	
	AddTaxonomySyncFolder (see page 363)	adds folder entry into taxonomy
	AddXmlConfiguration (see page 363)	Adds an XML configuration for XML oriented content.
	CheckOutContent (see page 363)	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 364)	Copy the content into required folder.
	CreateTaxonomy (see page 364)	creates new taxonomy
	DeleteContentItem (see page 364)	Delete the content by the content's ID.
	DeleteTaxonomy (see page 365)	deletes the existing taxonomy
	DeleteXmlConfiguration (see page 365)	Deletes an XML configuration.
	GetAddViewLanguage (see page 365)	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 365)	Loads all of the comments.
	GetAllTemplates (see page 366)	Loads all of the templates.

	GetAllUnassignedItemApprovals (see page 366)	Returns all of the unassigned user/groups for a given content ID.
	GetAllXmlConfigurations (see page 367)	Reads all of the XML configurations.
	GetChildContent (see page 367)	Loads all of the contents for the given folder.
	GetContent (see page 367)	Returns list of content. This function is overloaded.
	GetContentByHistory (see page 369)	Retrieves the corresponding content by the history ID.
	GetContentDisplayLink (see page 369)	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 370)	Loads the content details with checkout mode.
	GetContentReport (see page 370)	Loads the content data for the given constraints.
	GetContentState (see page 370)	Loads the content's details.
	GetContentStatus (see page 371)	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 371)	Gets web domain for a piece of content.
	GetExpireContent (see page 371)	Loads all of the expired content.
	GetTaxonomyIdByPath (see page 372)	returns taxonomy id by given path
	LoadTaxonomy (see page 372)	returns the TaxonomyData with all its subcategory and its items.
	MoveContentToFolder (see page 372)	Moves content to a folder.
	PublishContent (see page 373)	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.
	ReadAllAssignedCategory (see page 373)	returns all the assigned category for the content id
	ReadCategoryByTaxonomy (see page 374)	reads all the subcategory for given taxonomy
	ReadTaxonomy (see page 374)	read the taxonomy data
	RelicateXmlConfiguration (see page 374)	Relicate the XML configuration.
	RemoveTaxonomyFolder (see page 375)	removes the sync up folder from taxonomy
	RemoveTaxonomyItem (see page 375)	removes item from the taxonomy
	RenameContent (see page 375)	Renames the content.
	ReOrderTaxonomyItems (see page 375)	reorder display of taxonomy items
	RestoreHistoryContent (see page 376)	Restores the content to a particular version of content.
	SaveContent (see page 376)	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 377)	Delete the content by the content's Id and folder Id.
	UndoCheckout (see page 377)	This undoes the checkout for a piece of content which is checked out to the current user.
	UpdateContentMetaData (see page 377)	Updates a single piece of content metadata
	UpdateTaxonomy (see page 378)	updates the existing taxonomy
	UpdatexmlCollectionPackage (see page 378)	Updates the XML collection package.
	UpdateXmlConfiguration (see page 378)	Updates the XML configuration details.

Legend

	Method

Content Methods

The methods of the Content class are listed here.

Public Methods

	Name	Description
	AddContent (see page 360)	Adds Content to the CMS. This method is overloaded.
	AddTaxonomyItem (see page 362)	
	AddTaxonomySyncFolder (see page 363)	adds folder entry into taxonomy
	AddXmlConfiguration (see page 363)	Adds an XML configuration for XML oriented content (see page 354).
	CheckOutContent (see page 363)	This will check the content (see page 354) out to the current user if the content (see page 354) is currently checked in and the user has privileges to check it out.
	CopyContentToFolder (see page 364)	Copy the content (see page 354) into required folder.
	CreateTaxonomy (see page 364)	creates new taxonomy
	DeleteContentItem (see page 364)	Delete the content (see page 354) by the content (see page 354)'s ID.
	DeleteTaxonomy (see page 365)	deletes the existing taxonomy
	DeleteXmlConfiguration (see page 365)	Deletes an XML configuration.
	GetAddViewLanguage (see page 365)	Loads all the languages from the system for a given content (see page 354) ID and populates the Add/View drop down box with flag set on each item.
	GetAllComments (see page 365)	Loads all of the comments.
	GetAllTemplates (see page 366)	Loads all of the templates.
	GetAllUnassignedItemApprovals (see page 366)	Returns all of the unassigned user/groups for a given content (see page 354) ID.
	GetAllXmlConfigurations (see page 367)	Reads all of the XML configurations.
	GetChildContent (see page 367)	Loads all of the contents for the given folder.
	GetContent (see page 367)	Returns list of content. This function is overloaded.
	GetContentByHistory (see page 369)	Retrieves the corresponding content (see page 354) by the history ID.
	GetContentDisplayLink (see page 369)	Function takes in the required information about the string then will decide the best link to use to display the content (see page 354). It decides between Linkit, a generated quicklink or just the template file.
	GetContentForEditing (see page 370)	Loads the content (see page 354) details with checkout mode.
	GetContentReport (see page 370)	Loads the content (see page 354) data for the given constraints.

	GetContentState (see page 370)	Loads the content (see page 354)'s details.
	GetContentStatus (see page 371)	This will return a single letter string which is the status of the piece of content (see page 354). The letters returned are the same letters used in the Workarea view.
	GetDomain (see page 371)	Gets web domain for a piece of content (see page 354).
	GetExpireContent (see page 371)	Loads all of the expired content (see page 354).
	GetTaxonomyIdByPath (see page 372)	returns taxonomy id by given path
	LoadTaxonomy (see page 372)	returns the TaxonomyData with all its subcategory and its items.
	MoveContentToFolder (see page 372)	Moves content (see page 354) to a folder.
	PublishContent (see page 373)	Function takes in information about the content (see page 354) to be checked in and published. This goes through the normal approval chain. The content (see page 354) must be in a checked out state to the current logged in user for this to succeed.
	ReadAllAssignedCategory (see page 373)	returns all the assigned category for the content (see page 354) id
	ReadCategoryByTaxonomy (see page 374)	reads all the subcategory for given taxonomy
	ReadTaxonomy (see page 374)	read the taxonomy data
	RelicateXmlConfiguration (see page 374)	Relicate the XML configuration.
	RemoveTaxonomyFolder (see page 375)	removes the sync up folder from taxonomy
	RemoveTaxonomyItem (see page 375)	removes item from the taxonomy
	RenameContent (see page 375)	Renames the content (see page 354).
	ReOrderTaxonomyItems (see page 375)	reorder display of taxonomy items
	RestoreHistoryContent (see page 376)	Restores the content (see page 354) to a particular version of content (see page 354).
	SaveContent (see page 376)	Updates the checked out content (see page 354) 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 377)	Delete the content (see page 354) by the content (see page 354)'s Id and folder Id.
	UndoCheckout (see page 377)	This undoes the checkout for a piece of content (see page 354) which is checked out to the current user.
	UpdateContentMetaData (see page 377)	Updates a single piece of content (see page 354) metadata
	UpdateTaxonomy (see page 378)	updates the existing taxonomy
	UpdatexmlCollectionPackage (see page 378)	Updates the XML collection package.
	UpdateXmlConfiguration (see page 378)	Updates the XML configuration details.

Legend

	Method

AddContent Method

Adds Content to the CMS. This method is overloaded.

Overload List

	Name	Description
	Content.AddContent (String, String, String, String, String, String, Integer, Object, Object, Object) (see page 360)	Adds Content (see page 354) to the CMS
	Content.AddContent (String, String, String, String, String, String, Integer, Object, Object, Object, Integer) (see page 361)	Adds Content (see page 354) to the CMS
	Content.AddContent (String, String, String, String, String, String, Integer, Object, Object, Object, Integer, Integer) (see page 361)	Adds Content (see page 354) to the CMS

Content.AddContent Method (String, String, String, String, String, String, Integer, Object, Object, Object)

Adds Content (see page 354) to the CMS

C#

```
public int AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, int 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 Integer, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object) As Integer
```

Parameters

Parameters	Description
ContentTitle	Title of content (see page 354) to add
ContentComment	Content (see page 354) comments
ContentHtml	Content (see page 354) HTML to display
SearchText	Search Text
SummaryHtml	Summary HTML
ContentLanguage	Content (see page 354) Language
FolderId	Folder to place the content (see page 354) in
GoLive	Go Live Date
EndDate	Content (see page 354) End Date
MetaInfoXml	Meta Info XML

Returns

Integer - ID of newly created content (see page 354)

Content.AddContent Method (String, String, String, String, String, String, Integer, Object, Object, Object, Integer)

Adds Content (see page 354) to the CMS

C#

```
public int AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, int FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml, int 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 Integer, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object, ByVal ContentID As Integer) As Integer
```

Parameters

Parameters	Description
ContentTitle	Title of content (see page 354) to add
ContentComment	Content (see page 354) comments
ContentHtml	Content (see page 354) HTML to display
SearchText	Search Text
SummaryHtml	Summary HTML
ContentLanguage	Content (see page 354) Language
FolderId	Folder to place the content (see page 354) in
GoLive	Go Live Date
EndDate	Content (see page 354) End Date
MetaInfoXml	Meta Info XML
ContentID	ContentID

Returns

Integer - ID of newly created content (see page 354)

Content.AddContent Method (String, String, String, String, String, String, Integer, Object, Object, Object, Integer, Integer)

Adds Content (see page 354) to the CMS

C#

```
public int AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, int FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml, int XmlID, int 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 Integer, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object, ByVal XmlID As Integer, ByVal TemplateID As Integer) As
Integer
```

Parameters

Parameters	Description
ContentTitle	Title of content (see page 354) to add
ContentComment	Content (see page 354) comments

ContentHtml	Content (see page 354) HTML to display
SearchText	Search Text
SummaryHtml	Summary HTML
ContentLanguage	Content (see page 354) Language
FolderId	Folder to place the content (see page 354) in
GoLive	Go Live Date
EndDate	Content (see page 354) End Date
MetaInfoXml	Meta Info XML
XmlID	XmlID
TemplateID	Template ID

Returns

Integer - ID of newly created content (see page 354)

AddTaxonomyItem Method

Overload List

	Name	Description
	Content.AddTaxonomyItem (TaxonomyContentRequest) (see page 362)	inserts item into taxonomy
	Content.AddTaxonomyItem (TaxonomyRequest) (see page 362)	inserts item into taxonomy

Content.AddTaxonomyItem Method (TaxonomyContentRequest)

inserts item into taxonomy

C#

```
public AddTaxonomyItem(TaxonomyContentRequest request);
```

Visual Basic

```
Public Sub AddTaxonomyItem(ByVal request As TaxonomyContentRequest)
```

Parameters

Parameters	Description
request	taxonomycontentrequest

Content.AddTaxonomyItem Method (TaxonomyRequest)

inserts item into taxonomy

C#

```
public AddTaxonomyItem(TaxonomyRequest request);
```

Visual Basic

```
Public Sub AddTaxonomyItem(ByVal request As TaxonomyRequest)
```

Parameters

Parameters	Description
request	taxonomyrequest

Content.AddTaxonomySyncFolder Method

adds folder entry into taxonomy

C#

```
public AddTaxonomySyncFolder(TaxonomySyncRequest request);
```

Visual Basic

```
Public Sub AddTaxonomySyncFolder(ByVal request As TaxonomySyncRequest)
```

Parameters

Parameters	Description
request	taxonomyrequest

Content.AddXmlConfiguration Method

Adds an XML configuration for XML oriented content (see page 354).

C#

```
public int AddXmlConfiguration(Collection pagedata);
```

Visual Basic

```
Public Function AddXmlConfiguration(ByVal pagedata As Collection) As Integer
```

Parameters

Parameters	Description
pagedata	XML configuration data

Returns

Integer

Remarks

User login with administrator privileges are required to perform this operation.

Content.CheckOutContent Method

This will check the content (see page 354) out to the current user if the content (see page 354) is currently checked in and the user has privileges to check it out.

C#

```
public Boolean CheckOutContent(int ContentId);
```

Visual Basic

```
Public Function CheckOutContent(ByVal ContentId As Integer) As Boolean
```

Parameters

Parameters	Description
ContentId	Id of the content (see page 354) to check out

Returns

A Boolean representing the success of the check out

Remarks

User must be logged in and have sufficient privileges to check out the content (see page 354)

Content.CopyContentToFolder Method

Copy the content (see page 354) 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
Id	Content (see page 354) ID
FID	Folder ID
Lang	

Returns

Ektron.Cms.ContentData

Remarks

User login with administrator privileges are required to perform this operation.

Content.CreateTaxonomy Method

creates new taxonomy

C#

```
public int CreateTaxonomy(TaxonomyData request);
```

Visual Basic

```
Public Function CreateTaxonomy(ByVal request As TaxonomyData) As Integer
```

Parameters

Parameters	Description
request	taxonomy data

Returns

new taxonomy id

Content.DeleteContentItem Method

Delete the content (see page 354) by the content (see page 354)'s ID.

C#

```
public Boolean DeleteContentItem(int Id);
```

Visual Basic

```
Public Function DeleteContentItem(ByVal Id As Integer) As Boolean
```

Parameters

Parameters	Description
Id	The content (see page 354)'s ID

Returns

Boolean

Remarks

User login with administrator privileges are required to perform this operation.

Content.DeleteTaxonomy Method

deletes the existing taxonomy

C#

```
public DeleteTaxonomy(TaxonomyRequest request);
```

Visual Basic

```
Public Sub DeleteTaxonomy(ByVal request As TaxonomyRequest)
```

Parameters

Parameters	Description
request	taxonomy request

Content.DeleteXmlConfiguration Method

Deletes an XML configuration.

C#

```
public DeleteXmlConfiguration(int Id);
```

Visual Basic

```
Public Sub DeleteXmlConfiguration(ByVal Id As Integer)
```

Parameters

Parameters	Description
Id	XML configuration ID

Remarks

User login with administrator privileges are required to perform this operation.

Content.GetAddViewLanguage Method

Loads all the languages from the system for a given content (see page 354) ID and populates the Add/View drop down box with flag set on each item.

C#

```
public LanguageData GetAddViewLanguage(int Id);
```

Visual Basic

```
Public Function GetAddViewLanguage(ByVal Id As Integer) As LanguageData
```

Parameters

Parameters	Description
Id	Content (see page 354) ID

Returns

Ektron.Cms.LanguageData()

Content.GetAllComments Method

Loads all of the comments.

C#

```
public CommentData GetAllComments(int KeyId, int CommentId, int RefId, String RefType, int
UserId, String OrderBy);
```

Visual Basic

```
Public Function GetAllComments(ByVal KeyId As Integer, ByVal CommentId As Integer, ByVal RefId
As Integer, ByVal RefType As String, ByVal UserId As Integer, ByVal OrderBy As String) As
CommentData
```

Parameters

Parameters	Description
KeyId	key ID
CommentId	Comment ID
RefId	Reference ID, (content (see page 354) or forms)
RefType	Reference type, C = content (see page 354) or T = Task
UserId	The user ID
OrderBy	The order of the comments

Returns

Ektron.Cms.CommentData (see page 692)()

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
OrderBy	Order by clause

Returns

Ektron.Cms.TemplateData()

Remarks

User login required to perform this operation.

Content.GetAllUnassignedItemApprovals Method

Returns all of the unassigned user/groups for a given content (see page 354) ID.

C#

```
public ApprovalData GetAllUnassignedItemApprovals(int Id, String ItemType);
```

Visual Basic

```
Public Function GetAllUnassignedItemApprovals(ByVal Id As Integer, ByVal ItemType As String)
As ApprovalData
```

Parameters

Parameters	Description
Id	The content (see page 354)'s ID
ItemType	The type of user

Returns

Ektron.Cms.ApprovalData()

Content.GetAllXmlConfigurations Method

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
OrderBy	Order by clause

Returns

Ektron.Cms.XmlConfigData()

Remarks

User login required to perform this operation.

Content.GetChildContent Method

Loads all of the contents for the given folder.

C#

```
public ContentData GetChildContent(int Id, Boolean Recursive, String OrderBy);
```

Visual Basic

```
Public Function GetChildContent(ByVal Id As Integer, Optional ByVal Recursive As Boolean =
False, Optional ByVal OrderBy As String = "name") As ContentData
```

Parameters

Parameters	Description
Id	the folders ID
Recursive	True = recursive look up. False = none
OrderBy	An order by clause

Returns

Ektron.Cms.ContentData()

GetContent Method

Returns list of content. This function is overloaded.

Overload List

	Name	Description
	Content.GetContent (Ektron.Cms.Common.ContentRequest, System.Web.UI.Page) (see page 368)	Returns list of contents using content (see page 354) request object
	Content.GetContent (Integer, ContentResultType) (see page 368)	Loads content (see page 354) for viewers.

	Content.GetContent (Integer, Ektron.Cms.Content.EkContent.ContentResultType) (see page 368)	Loads the content (see page 354) details using the content (see page 354) ID

Content.GetContent Method (Ektron.Cms.Common.ContentRequest, System.Web.UI.Page)

Returns list of contents using content (see page 354) request object

C#

```
public Ektron.Cms.Common.ContentResult GetContent(Ektron.Cms.Common.ContentRequest req,
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
req	ContentRequest
page	Page reference

Returns

ContentResult

Content.GetContent Method (Integer, ContentResultType)

Loads content (see page 354) for viewers.

C#

```
public ContentData GetContent(int Id, ContentResultType ContentType);
```

Visual Basic

```
Public Function GetContent(ByVal Id As Integer, ByVal ContentType As ContentResultType) As
ContentData
```

Parameters

Parameters	Description
Id	The content (see page 354)'s ID
Preview	True or False flag

Returns

ContentData

Content.GetContent Method (Integer, Ektron.Cms.Content.EkContent.ContentResultType)

Loads the content (see page 354) details using the content (see page 354) ID

C#

```
public ContentData GetContent(int Id, Ektron.Cms.Content.EkContent.ContentResultType Type);
```

Visual Basic

```
Public Function GetContent(ByVal Id As Integer, Optional ByVal Type As
Ektron.Cms.Content.EkContent.ContentResultType = ContentResultType.Published) As ContentData
```

Parameters

Parameters	Description
Id	The content (see page 354) ID
Type	

Returns

Ektron.Cms.ContentData

Remarks

Read permission required.

Content.GetContentByHistory Method

Retrieves the corresponding content (see page 354) by the history ID.

C#

```
public ContentData GetContentByHistory(int Id);
```

Visual Basic

```
Public Function GetContentByHistory(ByVal Id As Integer) As ContentData
```

Parameters

Parameters	Description
Id	The history ID

Returns

Ektron.Cms.ContentData

Remarks

Read permission required.

Content.GetContentDisplayLink Method

Function takes in the required information about the string then will decide the best link to use to display the content (see page 354). 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
ContentType	Integer describing the content (see page 354)'s type (word doc, multimedia, etc.)
ContentId	A string value of the content (see page 354)'s ID number
ContentLanguage	A string value of the content (see page 354)'s language number
FolderId	A string value of the content (see page 354)'s folder ID number
host	A string value of the host name to build the link out of
StagingDomain	A string value of the content (see page 354)'s staging domain
ProductionDomain	A string value of the content (see page 354)'s production domain

Returns

The page link as a String

Content.GetContentForEditing Method

Loads the content (see page 354) details with checkout mode.

C#

```
public ContentEditData GetContentForEditing(int Id);
```

Visual Basic

```
Public Function GetContentForEditing(ByVal Id As Integer) As ContentEditData
```

Parameters

Parameters	Description
Id	The content (see page 354) ID.

Returns

Ektron.Cms.ContentEditData

Remarks

Edit permission required.

Content.GetContentReport Method

Loads the content (see page 354) 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
pagedata	required report data

Returns

Ektron.Cms.ContentData()

Remarks

User login with administrator privileges are required to perform this operation.

Content.GetContentState Method

Loads the content (see page 354)'s details.

C#

```
public ContentStateData GetContentState(int Id);
```

Visual Basic

```
Public Function GetContentState(ByVal Id As Integer) As ContentStateData
```

Parameters

Parameters	Description
Id	The content (see page 354)'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 (see page 354). The letters returned are the same letters used in the Workarea view.

C#

```
public String GetContentStatus(int ContentId);
```

Visual Basic

```
Public Function GetContentStatus(ByVal ContentId As Integer) As String
```

Parameters

Parameters	Description
ContentId	Id of the content (see page 354) to return the status of

Returns

A string with the single letter status of the content (see page 354)

Remarks

User must be logged in and have sufficient privileges to view the status

Content.GetDomain Method

Gets web domain for a piece of content (see page 354).

C#

```
public String GetDomain(int ContentId);
```

Visual Basic

```
Public Function GetDomain(ByVal ContentId As Integer) As String
```

Parameters

Parameters	Description
ContentId	The content (see page 354)'s ID

Returns

String

Content.GetExpireContent Method

Loads all of the expired content (see page 354).

C#

```
public ContentData GetExpireContent(Collection pagedata);
```

Visual Basic

```
Public Function GetExpireContent(ByVal pagedata As Collection) As ContentData
```

Parameters

Parameters	Description
pagedata	Required data for the report as a value, key pair

Returns

Ektron.Cms.ContentData()

Remarks

User login with administrator privileges are required to perform this operation.

Content.GetTaxonomyIdByPath Method

returns taxonomy id by given path

C#

```
public int GetTaxonomyIdByPath(String path);
```

Visual Basic

```
Public Function GetTaxonomyIdByPath(ByVal path As String) As Integer
```

Parameters

Parameters	Description
path	path

Returns

taxonomy id

Content.LoadTaxonomy Method

returns the TaxonomyData with all its subcategory and its items.

C#

```
public TaxonomyData LoadTaxonomy(TaxonomyRequest request);
```

Visual Basic

```
Public Function LoadTaxonomy(ByRef request As TaxonomyRequest) As TaxonomyData
```

Parameters

Parameters	Description
request	TaxonomyRequest

Returns

TaxonomyData

Content.MoveContentToFolder Method

Moves content (see page 354) 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
cId	The content (see page 354)'s ID
cLanguages	The content (see page 354)'s language
FID	The folder's ID

Returns

Ektron.Cms.ContentData

Remarks

User login with administrator privileges are required to perform this operation.

Content.PublishContent Method

Function takes in information about the content (see page 354) to be checked in and published. This goes through the normal approval chain. The content (see page 354) must be in a checked out state to the current logged in user for this to succeed.

C#

```
public Boolean PublishContent(int ContentId, int FolderId, int ContentLanguage, String
dontCreateTask, int UserID, String TaskTitle);
```

Visual Basic

```
Public Function PublishContent(ByVal ContentId As Integer, ByVal FolderId As Integer, ByVal
ContentLanguage As Integer, ByVal dontCreateTask As String, ByVal UserID As Integer, ByVal
TaskTitle As String) As Boolean
```

Parameters

Parameters	Description
ContentId	Integer value of the content (see page 354)'s ID
FolderId	Integer value of the content (see page 354) folder's ID
ContentLanguage	Integer value of the content (see page 354)'s language ID
dontCreateTask	If this parameter is "" a task will be created for this submit, else it will not
UserID	The user ID of task assigned by (current user). This is needed only if "dontCreateTask" is "", it is ignored else
TaskTitle	Title for the task. This is needed only if "dontCreateTask" is "", it is ignored else

Returns

A Boolean indicating success or not.

Remarks

User must be logged in and have sufficient privileges to submit content (see page 354) for publishing.

Content.ReadAllAssignedCategory Method

returns all the assigned category for the content (see page 354) id

C#

```
public TaxonomyBaseData ReadAllAssignedCategory(int contentid);
```

Visual Basic

```
Public Function ReadAllAssignedCategory(ByVal contentid As Integer) As TaxonomyBaseData
```

Parameters

Parameters	Description
contentid	contentid

Returns

array of taxonomybasedata

Content.ReadCategoryByTaxonomy Method

reads all the subcategory for given taxonomy

C#

```
public TaxonomyData ReadCategoryByTaxonomy(TaxonomyRequest request);
```

Visual Basic

```
Public Function ReadCategoryByTaxonomy(ByVal request As TaxonomyRequest) As TaxonomyData
```

Parameters

Parameters	Description
request	

Content.ReadTaxonomy Method

read the taxonomy data

C#

```
public TaxonomyData ReadTaxonomy(TaxonomyRequest request);
```

Visual Basic

```
Public Function ReadTaxonomy(ByVal request As TaxonomyRequest) As TaxonomyData
```

Parameters

Parameters	Description
request	taxonomyrequest

Returns

taxonomydata

Content.RelicateXmlConfiguration Method

Relicate the XML configuration.

C#

```
public int RelicateXmlConfiguration(int Id, String Title);
```

Visual Basic

```
Public Function RelicateXmlConfiguration(ByVal Id As Integer, ByVal Title As String) As Integer
```

Parameters

Parameters	Description
Id	The inherited XML ID
Title	Title

Returns

Integer

Remarks

User login with administrator privileges are required to perform this operation.

Content.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
request	taxonomyrequest

Content.RemoveTaxonomyItem Method

removes item from the taxonomy

C#

```
public RemoveTaxonomyItem(TaxonomyRequest request);
```

Visual Basic

```
Public Sub RemoveTaxonomyItem(ByVal request As TaxonomyRequest)
```

Parameters

Parameters	Description
request	taxonomyrequest

Content.RenameContent Method

Renames the content (see page 354).

C#

```
public ContentData RenameContent(int cId, String title);
```

Visual Basic

```
Public Function RenameContent(ByVal cId As Integer, ByVal title As String) As ContentData
```

Parameters

Parameters	Description
cId	The content (see page 354)'s ID
title	The content (see page 354)'s new title

Returns

Boolean

Remarks

User login with administrator privileges are required to perform this operation.

Content.ReOrderTaxonomyItems Method

reorder display of taxonomy items

C#

```
public ReOrderTaxonomyItems(TaxonomyRequest request);
```

Visual Basic

```
Public Sub ReOrderTaxonomyItems(ByVal request As TaxonomyRequest)
```

Parameters

Parameters	Description
request	

Content.RestoreHistoryContent Method

Restores the content (see page 354) to a particular version of content (see page 354).

C#

```
public RestoreHistoryContent(int Id);
```

Visual Basic

```
Public Sub RestoreHistoryContent(ByVal Id As Integer)
```

Parameters

Parameters	Description
Id	The content (see page 354)'s history ID

Remarks

User login with administrator privileges are required to perform this operation.

Content.SaveContent Method

Updates the checked out content (see page 354) 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
contentData	_nt_

Returns

False if successful

Remarks

Content (see page 354) must be in the checked out state.

Content.SubmitForDelete Method

Delete the content (see page 354) by the content (see page 354)'s Id and folder Id.

C#

```
public Boolean SubmitForDelete(int cId, int fId);
```

Visual Basic

```
Public Function SubmitForDelete(ByVal cId As Integer, ByVal fId As Integer) As Boolean
```

Parameters

Parameters	Description
cId	The content (see page 354)'s Id
fId	The content (see page 354)'s Folder Id

Returns

Boolean

Remarks

User login with delete permissions required.

Content.UndoCheckout Method

This undoes the checkout for a piece of content (see page 354) which is checked out to the current user.

C#

```
public Boolean UndoCheckout(int ContentId);
```

Visual Basic

```
Public Function UndoCheckout(ByVal ContentId As Integer) As Boolean
```

Parameters

Parameters	Description
ContentId	Id of the content (see page 354) for which to undo check out

Returns

A Boolean representing the success of the undo.

Remarks

User must be logged in and have sufficient privileges to undo the checkout.

Content.UpdateContentMetaData Method

Updates a single piece of content (see page 354) metadata

C#

```
public Boolean UpdateContentMetaData(int ContentID, int MetaID, String MetaText);
```

Visual Basic

```
Public Function UpdateContentMetaData(ByVal ContentID As Integer, ByVal MetaID As Integer,
ByVal MetaText As String) As Boolean
```

Parameters

Parameters	Description
ContentID	The Content (see page 354) ID
MetaID	The Metadata (see page 495) Type ID

MetaText	The new text to assign

Returns

boolean

Content.UpdateTaxonomy Method

updates the existing taxonomy

C#

```
public UpdateTaxonomy(TaxonomyData request);
```

Visual Basic

```
Public Sub UpdateTaxonomy(ByVal request As TaxonomyData)
```

Parameters

Parameters	Description
request	taxonomydata

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
package	The XML details

Remarks

User login with administrator privileges are required to perform this operation.

Content.UpdateXmlConfiguration Method

Updates the XML configuration details.

C#

```
public UpdateXmlConfiguration(Collection pagedata);
```

Visual Basic

```
Public Sub UpdateXmlConfiguration(ByVal pagedata As Collection)
```

Parameters

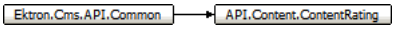
Parameters	Description
pagedata	The XML configuration details

Remarks

User login with administrator privileges are required to perform this operation.

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 API.Content.ContentRating.

Members

ContentRating Methods

	Name	Description
	GetContentRating (see page 380)	Constructs a string that represents a content (see page 337) rating element on a web site. This is composed of a rating dialog and a graph representing current ratings. Note that the content (see page 337) rating server control wraps this functionality.
	GetContentRatingAddRating (see page 381)	Adds a content (see page 337) rating to the database. Internally a check is done to ensure the user has not already rated the content (see page 337), and passes down the rating and any comments the user supplied.
	GetContentRatingResults (see page 381)	Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content (see page 337) 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 381)	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.
	PurgeContentRatings (see page 383)	Purges all ratings of a particular content (see page 337) block over a defined date/time range.

Topics

Name	Description
ContentRating Methods (see page 380)	The methods of the ContentRating class are listed here.

Legend

	Method

ContentRating Methods

	Name	Description
	GetContentRating (see page 380)	Constructs a string that represents a content (see page 337) rating element on a web site. This is composed of a rating dialog and a graph representing current ratings. Note that the content (see page 337) rating server control wraps this functionality.
	GetContentRatingAddRating (see page 381)	Adds a content (see page 337) rating to the database. Internally a check is done to ensure the user has not already rated the content (see page 337), and passes down the rating and any comments the user supplied.
	GetContentRatingResults (see page 381)	Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content (see page 337) 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 381)	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.
	PurgeContentRatings (see page 383)	Purges all ratings of a particular content (see page 337) block over a defined date/time range.

Legend

	Method

ContentRating Methods

The methods of the ContentRating class are listed here.

Public Methods

	Name	Description
	GetContentRating (see page 380)	Constructs a string that represents a content (see page 337) rating element on a web site. This is composed of a rating dialog and a graph representing current ratings. Note that the content (see page 337) rating server control wraps this functionality.
	GetContentRatingAddRating (see page 381)	Adds a content (see page 337) rating to the database. Internally a check is done to ensure the user has not already rated the content (see page 337), and passes down the rating and any comments the user supplied.
	GetContentRatingResults (see page 381)	Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content (see page 337) 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 381)	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.
	PurgeContentRatings (see page 383)	Purges all ratings of a particular content (see page 337) block over a defined date/time range.

Legend

	Method

ContentRating.GetContentRating Method

Constructs a string that represents a content (see page 337) rating element on a web site. This is composed of a rating dialog and a graph representing current ratings. Note that the content (see page 337) rating server control wraps this functionality.

C#

```
public String GetContentRating(int 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 Integer, 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
contentID	The numeric ID associated with the content (see page 337) block being rated.
currentPage	A reference to the System.Web.UI.Page object representing the active web page.
key	Specifies a unique ID of this element. No two content (see page 337) rating controls on the same page should specify the same key.
Labels	Specifies the labels of the control to be displayed. See Ektron.Cms.ContentRatingLabels
initialGraph	Specifies to show the initial (non-rated) graph to be horizontal to the control, below it, or not at all.
ratingCompleteGraph	Specifies to show the complete (rated) graph to be horizontal to the control, below it, or not at all.
padding	Specifies the number of pixels between the rating control and the graph.
bgColor	Specifies the background color of the graph.
barColor	Specifies the bar color of the graph.
TextColor	Specifies the text color of the graph.

Returns

Returns a string representing the final control.

Remarks

Note that the ContentRating (see page 379) server control nicely wraps this function.

ContentRating.GetContentRatingAddRating Method

Adds a content (see page 337) rating to the database. Internally a check is done to ensure the user has not already rated the content (see page 337), and passes down the rating and any comments the user supplied.

C#

```
public GetContentRatingAddRating(int contentid, int userid, int userRating, String
userComments);
```

Visual Basic

```
Public Sub GetContentRatingAddRating(ByVal contentid As Integer, ByVal userid As Integer,
ByVal userRating As Integer, ByVal userComments As String)
```

Parameters

Parameters	Description
contentid	The numeric ID associated with the content (see page 337) block being rated.
userid	The unique global identifier stored as a cookie by the user.
userRating	The rating the user supplies (1-9)
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 (see page 337) 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(int contentID, DateTime startDate, DateTime
endDate);
```

Visual Basic

```
Public Function GetContentRatingResults(ByVal contentID As Integer, ByVal startDate As
DateTime, ByVal endDate As DateTime) As System.Data.DataSet
```

Parameters

Parameters	Description
contentID	The numeric ID associated with the content (see page 337) block whose ratings are being queried.
startDate	The start date of a range of dates to query over.
endDate	The end date of a range of dates to query over.

Returns

(Username - String, RatingDate - DateTime, Rating - int, Comments - String)

GetContentRatingStatistics Method

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 (Integer, DateTime, DateTime) (see page 382)	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(Integer Integer String)
	ContentRating.GetContentRatingStatistics (Integer, Integer, String) (see page 382)	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(Integer DateTime DateTime)

ContentRating.GetContentRatingStatistics Method (Integer, 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(Integer Integer String)

C#

```
public Collection GetContentRatingStatistics(int contentID, DateTime startDate, DateTime
endDate);
```

Visual Basic

```
Public Function GetContentRatingStatistics(ByVal contentID As Integer, ByVal startDate As
DateTime, ByVal endDate As DateTime) As Collection
```

Parameters

Parameters	Description
contentID	The numeric ID associated with the content (see page 337) block being rated.
startDate	Bounds the result set by only ratings that occurred after this date.
endDate	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 (Integer, Integer, 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(Integer DateTime DateTime)

C#

```
public Collection GetContentRatingStatistics(int contentid, int user_id, String visitor_id);
```

Visual Basic

```
Public Function GetContentRatingStatistics(ByVal contentid As Integer, ByVal user_id As
Integer, ByVal visitor_id As String) As Collection
```

Parameters

Parameters	Description
contentid	The numeric ID associated with the content (see page 337) 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 (see page 337) block over a defined date/time range.

C#

```
public PurgeContentRatings(int contentid, DateTime startDate, DateTime endDate);
```

Visual Basic

```
Public Sub PurgeContentRatings(ByVal contentid As Integer, ByVal startDate As DateTime, ByVal
endDate As DateTime)
```

Parameters

Parameters	Description
contentid	The ID of the content (see page 337) that is having its ratings purged.
startDate	The beginning of the date range of content (see page 337) to be purged.
endDate	The end of the date range of content (see page 337) 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 API.Content.Form.

Members

Form Methods

	Name	Description
	AssignFormTask (see page 385)	Assigns a form submission task, as specified in FormInfo (see page 320), of the SubmittedDataXml.
	EmailFormFieldData (see page 385)	Sends an email, as specified in FormInfo (see page 320), of the SubmittedDataXml.
	GetAllForms (see page 386)	Retrieves an array of all the forms contained with the specified folder. This method is overloaded.
	GetForm (see page 387)	Retrieves information about the specified form.

	GetFormDataHistogram (see page 387)	
	GetFormFieldData (see page 387)	Retrieves the submitted data for all fields from a specified form. This method is overloaded.
	GetFormFieldList (see page 388)	Retrieves the list of form fields defined in the specified form.
	GetFormFieldQuestions (see page 389)	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 389)	Same as GetFormById, but returns only the form's title.
	SubmitFormFieldData (see page 389)	Adds form data as if it was submitted during a browser post to the web server.
	UpdateFormFieldData (see page 390)	Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

Topics

Name	Description
Form Methods (see page 384)	The methods of the Form class are listed here.

Legend

	Method

Form Methods

	Name	Description
	AssignFormTask (see page 385)	Assigns a form submission task, as specified in FormInfo (see page 320), of the SubmittedDataXml.
	EmailFormFieldData (see page 385)	Sends an email, as specified in FormInfo (see page 320), of the SubmittedDataXml.
	GetAllForms (see page 386)	Retrieves an array of all the forms contained with the specified folder. This method is overloaded.
	GetForm (see page 387)	Retrieves information about the specified form.
	GetFormDataHistogram (see page 387)	
	GetFormFieldData (see page 387)	Retrieves the submitted data for all fields from a specified form. This method is overloaded.
	GetFormFieldList (see page 388)	Retrieves the list of form fields defined in the specified form.
	GetFormFieldQuestions (see page 389)	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 389)	Same as GetFormById, but returns only the form's title.
	SubmitFormFieldData (see page 389)	Adds form data as if it was submitted during a browser post to the web server.
	UpdateFormFieldData (see page 390)	Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

Legend

	Method

Form Methods

The methods of the Form class are listed here.

Public Methods

	Name	Description
	AssignFormTask (see page 385)	Assigns a form submission task, as specified in FormInfo (see page 320), of the SubmittedDataXml.
	EmailFormFieldData (see page 385)	Sends an email, as specified in FormInfo (see page 320), of the SubmittedDataXml.
	GetAllForms (see page 386)	Retrieves an array of all the forms contained with the specified folder. This method is overloaded.
	GetForm (see page 387)	Retrieves information about the specified form.
	GetFormDataHistogram (see page 387)	

	GetFormFieldData (see page 387)	Retrieves the submitted data for all fields from a specified form. This method is overloaded.
	GetFormFieldList (see page 388)	Retrieves the list of form fields defined in the specified form.
	GetFormFieldQuestions (see page 389)	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 389)	Same as GetFormById, but returns only the form's title.
	SubmitFormFieldData (see page 389)	Adds form data as if it was submitted during a browser post to the web server.
	UpdateFormFieldData (see page 390)	Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

Legend

	Method

Form.AssignFormTask Method

Assigns a form submission task, as specified in FormInfo (see page 320), 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
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)
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 389) 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 389).

Form.EmailFormFieldData Method

Sends an email, as specified in FormInfo (see page 320), 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
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)
SubmittedDataXml	The form data submitted in XML format. Value Type: String (System.String)

Remarks

This routine is called automatically when calling SubmitFormFieldData (see page 389) 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 389).

GetAllForms Method

Retrieves an array of all the forms contained with the specified folder. This method is overloaded.

Overload List

	Name	Description
	Form.GetAllForms () (see page 386)	Returns all of the forms as an array of FormData.
	Form.GetAllForms (Integer, String, Integer, Cms.ContentAPI.FormResultType) (see page 386)	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()

Remarks

User login required to perform this operation.

Form.GetAllForms Method (Integer, 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(int Id, String OrderBy, int LanguageId,
Cms.ContentAPI.FormResultType Type);
```

Visual Basic

```
Public Function GetAllForms(ByVal Id As Integer, ByVal OrderBy As String, ByVal LanguageId As
Integer, ByVal Type As Cms.ContentAPI.FormResultType) As FormData
```

Parameters

Parameters	Description
Id	Specifies the folder.
OrderBy	Specifies how to sort the forms
LanguageId	Specifies a language or EkConstants.g_AllContentLanguages (-1) for all languages.
Type	Specifies whether only active forms or archived forms should be returned.

Remarks

Requires authentication.

Form.GetForm Method

Retrieves information about the specified form.

C#

```
public FormData GetForm(int Id);
```

Visual Basic

```
Public Function GetForm(ByVal Id As Integer) As FormData
```

Parameters

Parameters	Description
Id	Specifies the form.

Returns

Returns information about the form.

Remarks

Does not require authentication.

Form.GetFormDataHistogram Method

C#

```
public ArrayList GetFormDataHistogram(Long FormId);
```

Visual Basic

```
Public Function GetFormDataHistogram(ByVal FormId As Long) As ArrayList
```

Parameters

Parameters	Description
FormId	

GetFormFieldData Method

Retrieves the submitted data for all fields from a specified form. This method is overloaded.

Overload List

	Name	Description
	Form.GetFormFieldData (Integer, Integer) (see page 387)	Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.
	Form.GetFormFieldData (Integer, String, String) (see page 388)	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 (Integer, Integer)

Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

C#

```
public FormSubmittedData GetFormFieldData(int FormId, int FormDataId);
```

Visual Basic

```
Public Function GetFormFieldData(ByVal FormId As Integer, ByVal FormDataId As Integer) As
FormSubmittedData
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)
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 (Integer, 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(int FormId, String StartDate, String EndDate);
```

Visual Basic

```
Public Function GetFormFieldData(ByVal FormId As Integer, Optional ByVal StartDate As String =
"", Optional ByVal EndDate As String = "") As FormSubmittedData
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)
StartDate	Optionally limit the data returned to include only those submitted on or after the given date. Value Type: String (System.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.

C#

```
public FormFieldList GetFormFieldList(int FormId);
```

Visual Basic

```
Public Function GetFormFieldList(ByVal FormId As Integer) As FormFieldList
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)

Returns

The list for defined fields in the given form. Cms.FormFieldList (Ektron.Cms.FormFieldList)

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(int FormId);
```

Visual Basic

```
Public Function GetFormFieldQuestions(ByVal FormId As Integer) As Hashtable
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)

Returns

a table of defined fields with the corresponding questions in a given form.

Remarks

these questions is mainly used for reporting.

Form.GetFormTitle Method

Same as GetFormById, but returns only the form's title.

C#

```
public String GetFormTitle(int Id);
```

Visual Basic

```
Public Function GetFormTitle(ByVal Id As Integer) As String
```

Parameters

Parameters	Description
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.

C#

```
public SubmitFormFieldData(int FormId, FormSubmittedData Data, Boolean SendNotification,
[Out()] String SubmittedDataXml);
```

Visual Basic

```
Public Sub SubmitFormFieldData(ByVal FormId As Integer, ByVal Data As FormSubmittedData,
Optional ByVal SendNotification As Boolean = True, <Out()> Optional ByRef SubmittedDataXml As
String = "")
```

Parameters

Parameters	Description
FormId	Specifies the form that corresponds to the data. Value Type: Int32 (System.Int32)
Data	The data to submit. Value Type: Cms.FormSubmittedData (Ektron.Cms.FormSubmittedData)
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)
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.

C#

```
public UpdateFormFieldData(int FormId, int FormDataId, FormSubmittedData SubmittedData);
```

Visual Basic

```
Public Sub UpdateFormFieldData(ByVal FormId As Integer, ByVal FormDataId As Integer, ByVal
SubmittedData As FormSubmittedData)
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)
FormDataId	Specifies the instance of submitted data. Value Type: Int32 (System.Int32)
SubmittedData	The field data values to be updated. Value Type: Cms.FormSubmittedData (Ektron.Cms.FormSubmittedData)

Remarks

Does not require authentication.

ThreadedDiscussion Class

Class Hierarchy

C#

```
public class ThreadedDiscussion : Content;
```

Visual Basic

```
Public Class ThreadedDiscussion
Inherits Content
```

File

ThreadedDiscussion.vb

Description

This is class API.Content.ThreadedDiscussion.

Members

ThreadedDiscussion Methods

	Name	Description
	AddBoard (see page 393)	Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns integer ID of new discussion board.
	AddForum (see page 393)	Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns integer ID of new discussion forum.
	AddTopic (see page 394)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer. This method is overloaded.

	DeleteBoard (see page 395)	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.
	DeleteForum (see page 395)	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.
	DeleteTopic (see page 395)	Deletes a topic by topicID This method will delete all posts associated with the topic as well. Returns boolean value.
	GetBoardCategories (see page 396)	Method to get the Categories for a board by ID Returns either zero length or populated array. See DiscussionCategory class for more information.
	GetBoardForums (see page 396)	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 396)	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.
	GetDiscussionBoardString (see page 397)	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.
	GetForum (see page 398)	Method to obtain a particular forum by ID. Will return a discussion board with a 1 length Discussion Forum array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.
	GetRepliesForTopics (see page 398)	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 398)	Method to obtain a particular Topic by ID. Will return a discussion board with a 1 length Discussion Forum 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 399)	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.

Topics

Name	Description
ThreadedDiscussion Methods (see page 392)	The methods of the ThreadedDiscussion class are listed here.

Legend

	Method

ThreadedDiscussion Methods

	Name	Description
	AddBoard (see page 393)	Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns integer ID of new discussion board.
	AddForum (see page 393)	Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns integer ID of new discussion forum.
	AddTopic (see page 394)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer. This method is overloaded.
	DeleteBoard (see page 395)	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.
	DeleteForum (see page 395)	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.
	DeleteTopic (see page 395)	Deletes a topic by topicID This method will delete all posts associated with the topic as well. Returns boolean value.
	GetBoardCategories (see page 396)	Method to get the Categories for a board by ID Returns either zero length or populated array. See DiscussionCategory class for more information.
	GetBoardForums (see page 396)	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 396)	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.

	GetDiscussionBoardString (see page 397)	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.
	GetForum (see page 398)	Method to obtain a particular forum by ID. Will return a discussion board with a 1 length Discussion Forum array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.
	GetRepliesForTopics (see page 398)	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 398)	Method to obtain a particular Topic by ID. Will return a discussion board with a 1 length Discussion Forum 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 399)	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.

Legend

	Method

ThreadedDiscussion Methods

The methods of the ThreadedDiscussion class are listed here.

Public Methods

	Name	Description
	AddBoard (see page 393)	Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns integer ID of new discussion board.
	AddForum (see page 393)	Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns integer ID of new discussion forum.
	AddTopic (see page 394)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer. This method is overloaded.
	DeleteBoard (see page 395)	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.
	DeleteForum (see page 395)	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.
	DeleteTopic (see page 395)	Deletes a topic by topicID This method will delete all posts associated with the topic as well. Returns boolean value.
	GetBoardCategories (see page 396)	Method to get the Categories for a board by ID Returns either zero length or populated array. See DiscussionCategory class for more information.
	GetBoardForums (see page 396)	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 396)	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.
	GetDiscussionBoardString (see page 397)	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.
	GetForum (see page 398)	Method to obtain a particular forum by ID. Will return a discussion board with a 1 length Discussion Forum array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.

	GetRepliesForTopics (see page 398)	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 398)	Method to obtain a particular Topic by ID. Will return a discussion board with a 1 length Discussion Forum 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 399)	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.

Legend

	Method

ThreadedDiscussion.AddBoard Method

Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns integer ID of new discussion board.

C#

```
public int AddBoard(int ParentID, String Name, String Title, Boolean RequiresAuthentication,
String StyleSheet, String Categories, ));
```

Visual Basic

```
Public Function AddBoard(ByVal ParentID As Integer, ByVal Name As String, ByVal Title As
String, ByVal RequiresAuthentication As Boolean, ByVal StyleSheet As String, ByVal Categories
As String,  As )) As Integer
```

Parameters

Parameters	Description
ParentID	Integer ID of folder to add discussion board to.
Name	Name of board.
Title	Title of board.
RequiresAuthentication	Does the Board require authentication?
StyleSheet	The stylesheet to apply to the board.
Categories	String array of categories for the board.

Returns

Integer ID of new discussion board.

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 integer ID of new discussion forum.

C#

```
public int AddForum(int BoardID, String Name, String Description, Boolean ModeratePosts,
Boolean LockForum, int SortOrder, int CategoryID);
```

Visual Basic

```
Public Function AddForum(ByVal BoardID As Integer, ByVal Name As String, ByVal Description As
String, ByVal ModeratePosts As Boolean, ByVal LockForum As Boolean, ByVal SortOrder As
Integer, ByVal CategoryID As Integer) As Integer
```

Parameters

Parameters	Description
BoardID	
Name	
Description	String value for the forum summary
ModeratePosts	Are posts moderated?
LockForum	Is the forum locked?
SortOrder	Sort order for the new forum.
CategoryID	Integer of the Catagegory ID it belongs to.

Returns

Integer ID of new discussion forum.

AddTopic Method

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer. This method is overloaded.

Overload List

	Name	Description
	ThreadedDiscussion.AddTopic (Integer, String, String) (see page 394)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer.
	ThreadedDiscussion.AddTopic (Integer, String, String, Integer) (see page 394)	Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as integer.

ThreadedDiscussion.AddTopic Method (Integer, String, String)

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer.

C#

```
public int AddTopic(int ForumID, String Title, String Message);
```

Visual Basic

```
Public Function AddTopic(ByVal ForumID As Integer, ByVal Title As String, ByVal Message As
String) As Integer
```

Parameters

Parameters	Description
ForumID	Integer value of forum to add topic to.
Title	String value of the topic title.
Message	String value of the topic message.

Returns

Integer value for the topic ID.

ThreadedDiscussion.AddTopic Method (Integer, 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 integer.

C#

```
public int AddTopic(int ForumID, String Title, String Message, int LanguageId);
```

Visual Basic

```
Public Function AddTopic(ByVal ForumID As Integer, ByVal Title As String, ByVal Message As String, ByVal LanguageId As Integer) As Integer
```

Parameters

Parameters	Description
ForumID	Integer value of forum to add topic to.
Title	String value of the topic title.
Message	String value of the topic message.
LanguageId	Integer id of the Language ID

Returns

Integer value for the topic ID.

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.

C#

```
public DeleteBoard(int BoardID);
```

Visual Basic

```
Public Sub DeleteBoard(ByVal BoardID As Integer)
```

Parameters

Parameters	Description
BoardID	The integer 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.

C#

```
public DeleteForum(int forumID);
```

Visual Basic

```
Public Sub DeleteForum(ByVal forumID As Integer)
```

Parameters

Parameters	Description
forumID	The integer ID of the forum. This corresponds to the folder id.

ThreadedDiscussion.DeleteTopic Method

Deletes a topic by topicID This method will delete all posts associated with the topic as well. Returns boolean value.

C#

```
public Boolean DeleteTopic(int topicid);
```

Visual Basic

```
Public Function DeleteTopic(ByVal topicid As Integer) As Boolean
```

Parameters

Parameters	Description
post_id	Integer 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 class for more information.

C#

```
public DiscussionCategory GetBoardCategories(int boardID);
```

Visual Basic

```
Public Function GetBoardCategories(ByVal boardID As Integer) As DiscussionCategory
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.

Returns

Returns an array of type DiscussionCategory.

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(int boardID);
```

Visual Basic

```
Public Function GetBoardForums(ByVal boardID As Integer) As DiscussionForum
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.

Returns

Returns an array of type DiscussionForum.

GetDiscussionBoard Method

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 (Integer) (see page 397)	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 397)	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.

ThreadedDiscussion.GetDiscussionBoard Method (Integer)

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(int boardID);
```

Visual Basic

```
Public Function GetDiscussionBoard(ByVal boardID As Integer) As DiscussionBoard
```

Parameters

Parameters	Description
boardID	Integer 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.

C#

```
public DiscussionBoard GetDiscussionBoard(String boardpath);
```

Visual Basic

```
Public Function GetDiscussionBoard(ByVal boardpath As String) As DiscussionBoard
```

Parameters

Parameters	Description
boardpath	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 CSS stylesheets.

C#

```
public String GetDiscussionBoardString(int boardID, String URLpath);
```

Visual Basic

```
Public Function GetDiscussionBoardString(ByVal boardID As Integer, ByVal URLpath As String) As
String
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.

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 array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.

C#

```
public DiscussionBoard GetForum(int forumid);
```

Visual Basic

```
Public Function GetForum(ByVal forumid As Integer) As DiscussionBoard
```

Parameters

Parameters	Description
forumid	Integer 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

C#

```
public ArrayList GetRepliesForTopics(int ForumID);
```

Visual Basic

```
Public Function GetRepliesForTopics(ByVal ForumID As Integer) As ArrayList
```

Parameters

Parameters	Description
ForumID	Integer 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 398)(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 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(int topicid);
```

Visual Basic

```
Public Function GetTopic(ByVal topicid As Integer) As DiscussionBoard
```

Parameters

Parameters	Description
topicid	Integer 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.

C#

```
public EkTasks SearchReplies(int ForumIDs, ), String SearchTerms, ), EkEnumeration.SearchTypes
stType);
```

Visual Basic

```
Public Function SearchReplies(ByVal ForumIDs As Integer,  As ), ByVal SearchTerms As String,
As ), ByVal stType As EkEnumeration.SearchTypes) As EkTasks
```

Parameters

Parameters	Description
ForumIDs	Integer array of the Forum IDs you want to search. Use zero length array, or 0, to search all.
SearchTerms	String array of the search terms (case insensitive).
stType	EkEnumeration.SearchTypes enumeration value for the search type.

Returns

An instance of type ekTasks (an array of Tasks).

API.Search Namespace

This is namespace API.Search.

Classes

	Name	Description
	SearchManager (see page 400)	This is class API.Search.SearchManager.

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

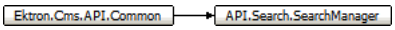
	Name	Description
	SearchManager (see page 400)	This is class API.Search.SearchManager.

Legend

	Class

SearchManager Class

Class Hierarchy



C#

```
public class SearchManager : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class SearchManager
Inherits Ektron.Cms.API.Common
```

File

SearchManager.vb

Description

This is class API.Search.SearchManager.

Members

SearchManager Methods

	Name	Description
	Execute (see page 401)	Returns the array of ContentData. This method is overloaded.

Topics

Name	Description
SearchManager Methods (see page 400)	The methods of the SearchManager class are listed here.

Legend

	Method
	virtual

SearchManager Methods

	Name	Description
	Execute (see page 401)	Returns the array of ContentData. This method is overloaded.

Legend

	Method
	virtual

SearchManager Methods

The methods of the SearchManager class are listed here.

Public Methods

	Name	Description
	Execute (see page 401)	Returns the array of ContentData. This method is overloaded.

Legend

	Method
	virtual

Execute Method

Returns the array of ContentData. This method is overloaded.

Overload List

	Name	Description
	SearchManager.Execute (Ektron.Cms.ContentSearchCondition) (see page 401)	Returns the array of ContentData based on the criteria ContentSearchCondition.
	SearchManager.Execute (Ektron.Cms.UserSearchCondition) (see page 402)	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
condition	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
condition	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)
```

API.User Namespace

This is namespace API.User.

Classes

	Name	Description
	ADUser (see page 403)	This is class API.User.ADUser.
	User (see page 413)	This is class API.User.User.

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

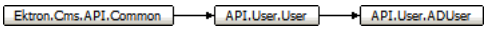
	Name	Description
	ADUser (see page 403)	This is class API.User.ADUser.
	User (see page 413)	This is class API.User.User.

Legend

	Class

ADUser Class

Class Hierarchy



C#

```
public class ADUser : User;
```

Visual Basic

```
Public Class ADUser
Inherits User
```

File

ADUser.vb

Description

This is class API.User.ADUser.

Members

ADUser Methods

	Name	Description
	AddADGroupToCMS (see page 405)	This function allows you add Active Directory groups to the CMS.
	AddADUsersToCMSByUsername (see page 406)	Adds Active Directory users to the CMS by their user name and domain.
	DeSynchUserGroups (see page 406)	Updates the group with a new group name.
	DeSynchUsers (see page 406)	Update the user with new user details.
	GetADMapping (see page 407)	Loads the CMS and Active Directory name.
	GetADStatus (see page 407)	Used to load required Synch, Desynch and relationship flags of the configured Active Directory.
	GetAvailableADGroups (see page 407)	Loads all of the Active Directory groups by a given domain and filter.
	GetAvailableADUsers (see page 408)	Loads all available Active Directory users.
	GetCMSGroupsToSync (see page 408)	Loads all the CMS groups that need to be synchronized.
	GetCMSRelationshipsToSync (see page 409)	Loads all the CMS relationships that need to be synchronized with Active Directory.
	GetCMSUsersToSync (see page 409)	Loads all the CMS users that need to be synchronized with Active Directory.
	GetDomains (see page 409)	Loads a list of domains from Active Directory.
	MapCMSUserGroupToAD (see page 410)	Updates the given CMS group corresponding to the Active Directory group.
	MapCMSUserToAD (see page 410)	Maps the given user ID to the corresponding Active Directory entry.
	SynchCMSGroupsToAD (see page 410)	Synchronizes the CMS group data to the Active Directory group data. This updates the CMS data with the corresponding to Active Directory changes.
	SynchCMSRelationshipsToAD (see page 411)	Synchronizes the CMS relations, such as groups, to the Active Directory for the given users.

	SynchCMSUsersToAD (see page 411)	Synchronize the CMS user data to the Active Directory data.
	UpdateADMapping (see page 412)	Updates the Active directory information into the CMS based on the user ID and user data.
	UpdateADUsersGroups (see page 412)	Updates Active Directory users in a CMS group.
	UpdateUsersInfoFromAD (see page 412)	Updates the user details from the Active Directory details.

Topics

Name	Description
ADUser Methods (see page 404)	The methods of the ADUser class are listed here.

Legend

	Method

ADUser Methods

	Name	Description
	AddADGroupToCMS (see page 405)	This function allows you add Active Directory groups to the CMS.
	AddADUsersToCMSByUsername (see page 406)	Adds Active Directory users to the CMS by their user name and domain.
	DeSynchUserGroups (see page 406)	Updates the group with a new group name.
	DeSynchUsers (see page 406)	Update the user with new user details.
	GetADMapping (see page 407)	Loads the CMS and Active Directory name.
	GetADStatus (see page 407)	Used to load required Synch, Desynch and relationship flags of the configured Active Directory.
	GetAvailableADGroups (see page 407)	Loads all of the Active Directory groups by a given domain and filter.
	GetAvailableADUsers (see page 408)	Loads all available Active Directory users.
	GetCMSGroupsToSync (see page 408)	Loads all the CMS groups that need to be synchronized.
	GetCMSRelationshipsToSync (see page 409)	Loads all the CMS relationships that need to be synchronized with Active Directory.
	GetCMSUsersToSync (see page 409)	Loads all the CMS users that need to be synchronized with Active Directory.
	GetDomains (see page 409)	Loads a list of domains from Active Directory.
	MapCMSUserGroupToAD (see page 410)	Updates the given CMS group corresponding to the Active Directory group.
	MapCMSUserToAD (see page 410)	Maps the given user ID to the corresponding Active Directory entry.
	SynchCMSGroupsToAD (see page 410)	Synchronizes the CMS group data to the Active Directory group data. This updates the CMS data with the corresponding to Active Directory changes.
	SynchCMSRelationshipsToAD (see page 411)	Synchronizes the CMS relations, such as groups, to the Active Directory for the given users.
	SynchCMSUsersToAD (see page 411)	Synchronize the CMS user data to the Active Directory data.
	UpdateADMapping (see page 412)	Updates the Active directory information into the CMS based on the user ID and user data.
	UpdateADUsersGroups (see page 412)	Updates Active Directory users in a CMS group.
	UpdateUsersInfoFromAD (see page 412)	Updates the user details from the Active Directory details.

Legend

	Method

ADUser Methods

The methods of the ADUser class are listed here.

Public Methods

	Name	Description
	AddADGroupToCMS (see page 405)	This function allows you add Active Directory groups to the CMS.
	AddADUsersToCMSByUsername (see page 406)	Adds Active Directory users to the CMS by their user name and domain.
	DeSynchUserGroups (see page 406)	Updates the group with a new group name.
	DeSynchUsers (see page 406)	Update the user with new user details.
	GetADMapping (see page 407)	Loads the CMS and Active Directory name.
	GetADStatus (see page 407)	Used to load required Synch, Desynch and relationship flags of the configured Active Directory.
	GetAvailableADGroups (see page 407)	Loads all of the Active Directory groups by a given domain and filter.
	GetAvailableADUsers (see page 408)	Loads all available Active Directory users.
	GetCMSGroupsToSync (see page 408)	Loads all the CMS groups that need to be synchronized.
	GetCMSRelationshipsToSync (see page 409)	Loads all the CMS relationships that need to be synchronized with Active Directory.
	GetCMSUsersToSync (see page 409)	Loads all the CMS users that need to be synchronized with Active Directory.
	GetDomains (see page 409)	Loads a list of domains from Active Directory.
	MapCMSUserGroupToAD (see page 410)	Updates the given CMS group corresponding to the Active Directory group.
	MapCMSUserToAD (see page 410)	Maps the given user ID to the corresponding Active Directory entry.
	SynchCMSGroupsToAD (see page 410)	Synchronizes the CMS group data to the Active Directory group data. This updates the CMS data with the corresponding to Active Directory changes.
	SynchCMSRelationshipsToAD (see page 411)	Synchronizes the CMS relations, such as groups, to the Active Directory for the given users.
	SynchCMSUsersToAD (see page 411)	Synchronize the CMS user data to the Active Directory data.
	UpdateADMapping (see page 412)	Updates the Active directory information into the CMS based on the user ID and user data.
	UpdateADUsersGroups (see page 412)	Updates Active Directory users in a CMS group.
	UpdateUsersInfoFromAD (see page 412)	Updates the user details from the Active Directory details.

Legend

	Method

ADUser.AddADGroupToCMS Method

This function allows you add Active Directory groups to the CMS.

C#

```
public AddADGroupToCMS(Object GroupPaths);
```

Visual Basic

```
Public Sub AddADGroupToCMS(ByVal GroupPaths As Object)
```

Parameters

Parameters	Description
GroupPaths	A collection of Active Directory group paths.

Remarks

A user with administrator privileges must be logged in to perform this operation.

ADUser.AddADUsersToCMSByUsername Method

Adds Active Directory 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
UserNames	A collection of user names.
UserDomains	A collection of domain names.

Remarks

A user with administrator privileges must be logged in to perform this operation.

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
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.

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
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.

ADUser.GetADMapping Method

Loads the CMS and Active Directory name.

C#

```
public AdMappingData GetADMapping(int CallerId, String MapType, int Map, int Exclude, int
Editable);
```

Visual Basic

```
Public Function GetADMapping(ByVal CallerId As Integer, ByVal MapType As String, ByVal Map As
Integer, ByVal Exclude As Integer, ByVal Editable As Integer) As AdMappingData
```

Parameters

Parameters	Description
`CallerId`	Who is calling this function.
`MapType`	Possible value is “userprop” (user properties).
`Map`	Ektron.Cms.Common.EkEnumeration.MapTypes (include, exclude and dontconsider are the possible values).
`Exclude`	Flag (0 – false, 1-true).
`Editable`	Flag(0-false,1-true).

Returns

Ektron.Cms.AdMappingData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

ADUser.GetADStatus Method

Used to load required Synch, Desynch and relationship flags of the configured Active Directory.

C#

```
public AdSyncData GetADStatus();
```

Visual Basic

```
Public Function GetADStatus() As AdSyncData
```

Returns

Ektron.Cms.AdSyncData

Remarks

A user with administrator privileges must be logged in to perform this operation.

ADUser.GetAvailableADGroups Method

Loads all of the Active Directory groups by 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
Filter	Filter by complete or a specific group name.
Domain	The domain name. (Required)

Returns

Ektron.Cms.GroupData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

ADUser.GetAvailableADUsers Method

Loads all available Active Directory 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
Attributes	Possible attributes are firstname, lastname, username or any(*).
Filter	Possible filters are firstname, lastname, username or any(*).
OrderBy	Any specific order such as username, firstname, lastname and domain.
Domain	The active directory domain. (Required)

Returns

Ektron.Cms.UserData()

Remarks

A user with administrator privileges must be logged in to perform this operation.

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
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.

ADUser.GetCMSRelationshipsToSync Method

Loads all the CMS relationships that need to be synchronized with Active Directory.

C#

```
public UserGroupData GetCMSRelationshipsToSync(int MaxReturn);
```

Visual Basic

```
Public Function GetCMSRelationshipsToSync(ByVal MaxReturn As Integer) As UserGroupData
```

Parameters

Parameters	Description
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.

ADUser.GetCMSUsersToSync Method

Loads all the CMS users that need to be synchronized with Active Directory.

C#

```
public UserData GetCMSUsersToSync(int MaxReturn);
```

Visual Basic

```
Public Function GetCMSUsersToSync(ByVal MaxReturn As Integer) As UserData
```

Parameters

Parameters	Description
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.

ADUser.GetDomains Method

Loads a list of domains from Active Directory.

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
ShowAll	flag(1 = show all).
CheckIntegration	Check whether CMS is configured to integrate Active Directory (1 = check otherwise resumed).

Returns

Ektron.Cms.DomainData()

ADUser.MapCMSUserGroupToAD Method

Updates the given CMS group corresponding to the Active Directory group.

C#

```
public MapCMSUserGroupToAD(int GroupId, String GroupName, String Domain);
```

Visual Basic

```
Public Sub MapCMSUserGroupToAD(ByVal GroupId As Integer, ByVal GroupName As String, ByVal
Domain As String)
```

Parameters

Parameters	Description
GroupId	CMS group ID.
GroupName	CMS group name.
Domain	The Active Directory domain.

Remarks

A user with administrator privileges must be logged in to perform this operation.

ADUser.MapCMSUserToAD Method

Maps the given user ID to the corresponding Active Directory entry.

C#

```
public MapCMSUserToAD(int UserId, String UserName, String Domain, int PullInfo);
```

Visual Basic

```
Public Sub MapCMSUserToAD(ByVal UserId As Integer, ByVal UserName As String, ByVal Domain As
String, ByVal PullInfo As Integer)
```

Parameters

Parameters	Description
UserId	The user's ID.
UserName	The user's name.
Domain	The user's domain.
PullInfo	Flag to load Active Directory information. (1 = Load Active Directory information, otherwise none.)

Remarks

A user with administrator privileges must be logged in to perform this operation.

ADUser.SynchCMSGroupsToAD Method

Synchronizes the CMS group data to the Active Directory group data. This updates the CMS data with the corresponding to Active Directory 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
GroupIds	A collection of group IDs.
UserNames	A collection of usernames.
Domains	A collection of domains.
Actions	Possible actions are “delete” or “map”.

Remarks

A user with administrator privileges must be logged in to perform this operation.

ADUser.SynchCMSRelationshipsToAD Method

Synchronizes the CMS relations, such as groups, to the Active Directory 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
UserIds	A collection of user IDs.
GroupIds	A collection of group IDs.
Actions	A collection of actions (possible action is “delete”).

Remarks

A user with administrator privileges must be logged in to perform this operation.

ADUser.SynchCMSUsersToAD Method

Synchronize the CMS user data to the Active Directory 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
UserIds	A collection of user IDs.
UserNames	A collection of usernames.
Domains	A collection of domains.
Actions	A collection of actions corresponding to items added to the collection. The possible collection values are “delete” or “map”.
PullInfo	Retrieves the Active Directory information when the value is 1, otherwise none.

Remarks

A user with administrator privileges must be logged in to perform this operation.

ADUser.UpdateADMapping Method

Updates the Active directory information into the CMS based on the user ID and user data.

C#

```
public UpdateADMapping(int UserId, Object pagedata);
```

Visual Basic

```
Public Sub UpdateADMapping(ByVal UserId As Integer, ByVal pagedata As Object)
```

Parameters

Parameters	Description
UserId	The user's ID.
pagedata	Key, value pair for given user ID.

Remarks

A user with administrator privileges must be logged in to perform this operation.

ADUser.UpdateADUsersGroups Method

Updates Active Directory 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
UserName	The user's username.
Domain	The user's Domain.

Remarks

A user with administrator privileges must be logged in to perform this operation.

ADUser.UpdateUsersInfoFromAD Method

Updates the user details from the Active Directory details.

C#

```
public UpdateUsersInfoFromAD(String UserName, String Domain);
```

Visual Basic

```
Public Sub UpdateUsersInfoFromAD(ByVal UserName As String, ByVal Domain As String)
```

Parameters

Parameters	Description
UserName	The user's username.
Domain	The user's Domain.

Remarks

A user with administrator privileges must be logged in to perform this operation.

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 API.User.User.

Members

User Methods

	Name	Description
	ActivateUserAccount (see page 418)	Activates a deleted (deactivated) user into the current/active state using the UserId or the Username and GUID (AccountID) parameters. This method is overloaded
	AddMembershipUser (see page 419)	Adds a new membership user to CMS.
	AddNewUser (see page 420)	Adds a new user to CMS.
	AddRolePermission (see page 420)	Adds permission for the given role.
	AddUser (see page 420)	
	AddUserGroup (see page 421)	Creates new user group.
	AddUserToGroup (see page 421)	Adds an existing user to an existing group.
	AutoLogInUser (see page 422)	Login the given user credentials.
	CreateBlankUserObject (see page 422)	This method is overloaded.
	DeleteUserByID (see page 423)	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 423)	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 423)	Removes a user from the designated group.
	DeleteUserGroup (see page 424)	Removes the group from the system permanently.
	EditUserCustomProperties (see page 424)	This method creates the editable fields(UI) for user's custom properties.
	GetActiveUser (see page 424)	Returns all of the user properties with or without preference.
	GetADMapping (see page 424)	Loads the CMS and Active Directory name.
	GetAllCmsUsers (see page 425)	Loads all the CMS users.
	GetAllCustomProperties (see page 425)	Returns array of all the custom properties defined in the system.
	GetAllUserByCustomProperty (see page 426)	Returns all the users based on the given properties.

	GetAllUserGroups (see page 426)	Loads all of the user groups from the system. This method is overloaded.
	GetAllUsers (see page 427)	This method is overloaded.
	GetGroupsUserIsIn (see page 428)	Loads all the groups by user.
	GetMembershipUser (see page 428)	Returns user data based on the user's login name (username).
	GetUser (see page 428)	Loads the user details for the given user.
	GetUserByUsername (see page 429)	Loads the user details for the given user.
	GetUserCustomProperties (see page 429)	Returns an array of custom properties for a given user's ID.
	GetUserEmailInfo (see page 429)	Returns user data for a given user's ID.
	GetUserForFolderAdmin (see page 430)	
	GetUserGroup (see page 430)	Loads a given group's details.
	GetUserGroupByName (see page 430)	Loads a given group's details.
	GetUserGroupForFolderAdmin (see page 431)	Returns all the group's data for the administrator.
	GetUserPermissions (see page 431)	Loads the user permissions as a array of UserPermissionData.
	GetUserPreference (see page 431)	Loads the preferences for the given user.
	GetUsers (see page 432)	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.
	GetUsersForTask (see page 433)	Loads the users that have permissions to the given content (see page 337).
	GetUsersNotInGroup (see page 433)	Loads all the users that are not in the given group.
	IsAdmin (see page 434)	Determines the if the current user is an Administrator.
	IsARoleMember (see page 434)	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.
	IsARoleMemberForContent (see page 436)	Determines whether a Role Member is allowed to work with a given piece of content (see page 337) as an administrator or not. Administrations privileges are for the given piece of content (see page 337) 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 437)	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 437)	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 438)	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 438)	Determines whether a Role Member is allowed to work with a given folder as an meta data administrator or not.
	LogInUser (see page 438)	Used to log in the user.
	ReadCustomProperties (see page 439)	Returns the custom properties as a Visual Basic collection. Importing the Microsoft.VisualBasic name space might be required.
	ResetMembershipUserPassword (see page 439)	Reset the password for the given membership user.
	SetAuthenticationCookie (see page 439)	This method sets the authentication cookie.
	UnsubscribeUser (see page 440)	Unsubscribes a user from the system.
	UpdateCustomPropertiesItemOrder (see page 440)	Updates the order of available custom properties.
	UpdateUser (see page 441)	
	UpDateUserGroup (see page 441)	
	UpdateUserPreferences (see page 442)	Updates the given user's preference data.

	UserObject (see page 442)	Returns user data based on the user's ID or by the user's login or username. This function is overloaded.

Topics

Name	Description
User Methods (see page 416)	The methods of the User class are listed here.

Legend

	Method

User Methods

	Name	Description
	ActivateUserAccount (see page 418)	Activates a deleted (deactivated) user into the current/active state using the UserId or the Username and GUID (AccountID) parameters. This method is overloaded
	AddMembershipUser (see page 419)	Adds a new membership user to CMS.
	AddNewUser (see page 420)	Adds a new user to CMS.
	AddRolePermission (see page 420)	Adds permission for the given role.
	AddUser (see page 420)	
	AddUserGroup (see page 421)	Creates new user group.
	AddUserToGroup (see page 421)	Adds an existing user to an existing group.
	AutoLogInUser (see page 422)	Login the given user credentials.
	CreateBlankUserObject (see page 422)	This method is overloaded.
	DeleteUserByID (see page 423)	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 423)	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 423)	Removes a user from the designated group.
	DeleteUserGroup (see page 424)	Removes the group from the system permanently.
	EditUserCustomProperties (see page 424)	This method creates the editable fields(UI) for user's custom properties.
	GetActiveUser (see page 424)	Returns all of the user properties with or without preference.
	GetADMapping (see page 424)	Loads the CMS and Active Directory name.
	GetAllCmsUsers (see page 425)	Loads all the CMS users.
	GetAllCustomProperties (see page 425)	Returns array of all the custom properties defined in the system.
	GetAllUserByCustomProperty (see page 426)	Returns all the users based on the given properties.
	GetAllUserGroups (see page 426)	Loads all of the user groups from the system. This method is overloaded.
	GetAllUsers (see page 427)	This method is overloaded.
	GetGroupsUserIsIn (see page 428)	Loads all the groups by user.
	GetMembershipUser (see page 428)	Returns user data based on the user's login name (username).
	GetUser (see page 428)	Loads the user details for the given user.
	GetUserByUsername (see page 429)	Loads the user details for the given user.
	GetUserCustomProperties (see page 429)	Returns an array of custom properties for a given user's ID.

	GetUserEmailInfo (see page 429)	Returns user data for a given user's ID.
	GetUserForFolderAdmin (see page 430)	
	GetUserGroup (see page 430)	Loads a given group's details.
	GetUserGroupByName (see page 430)	Loads a given group's details.
	GetUserGroupForFolderAdmin (see page 431)	Returns all the group's data for the administrator.
	GetUserPermissions (see page 431)	Loads the user permissions as a array of UserPermissionData.
	GetUserPreference (see page 431)	Loads the preferences for the given user.
	GetUsers (see page 432)	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.
	GetUsersForTask (see page 433)	Loads the users that have permissions to the given content (see page 337).
	GetUsersNotInGroup (see page 433)	Loads all the users that are not in the given group.
	IsAdmin (see page 434)	Determines the if the current user is an Administrator.
	IsARoleMember (see page 434)	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.
	IsARoleMemberForContent (see page 436)	Determines whether a Role Member is allowed to work with a given piece of content (see page 337) as an administrator or not. Administrations privileges are for the given piece of content (see page 337) 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 437)	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 437)	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 438)	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 438)	Determines whether a Role Member is allowed to work with a given folder as an meta data administrator or not.
	LogInUser (see page 438)	Used to log in the user.
	ReadCustomProperties (see page 439)	Returns the custom properties as a Visual Basic collection. Importing the Microsoft.VisualBasic name space might be required.
	ResetMembershipUserPassword (see page 439)	Reset the password for the given membership user.
	SetAuthenticationCookie (see page 439)	This method sets the authentication cookie.
	UnsubscribeUser (see page 440)	Unsubscribes a user from the system.
	UpdateCustomPropertiesItemOrder (see page 440)	Updates the order of available custom properties.
	UpdateUser (see page 441)	
	UpDateUserGroup (see page 441)	
	UpdateUserPreferences (see page 442)	Updates the given user's preference data.
	UserObject (see page 442)	Returns user data based on the user's ID or by the user's login or username. This function is overloaded.

Legend

	Method

User Methods

The methods of the User class are listed here.

Public Methods

	Name	Description
	ActivateUserAccount (see page 418)	Activates a deleted (deactivated) user into the current/active state using the UserId or the Username and GUID (AccountID) parameters. This method is overloaded
	AddMembershipUser (see page 419)	Adds a new membership user to CMS.
	AddNewUser (see page 420)	Adds a new user to CMS.
	AddRolePermission (see page 420)	Adds permission for the given role.
	AddUser (see page 420)	
	AddUserGroup (see page 421)	Creates new user group.
	AddUserToGroup (see page 421)	Adds an existing user to an existing group.
	AutoLogInUser (see page 422)	Login the given user credentials.
	CreateBlankUserObject (see page 422)	This method is overloaded.
	DeleteUserByID (see page 423)	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 423)	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 423)	Removes a user from the designated group.
	DeleteUserGroup (see page 424)	Removes the group from the system permanently.
	EditUserCustomProperties (see page 424)	This method creates the editable fields(UI) for user's custom properties.
	GetActiveUser (see page 424)	Returns all of the user properties with or without preference.
	GetADMapping (see page 424)	Loads the CMS and Active Directory name.
	GetAllCmsUsers (see page 425)	Loads all the CMS users.
	GetAllCustomProperties (see page 425)	Returns array of all the custom properties defined in the system.
	GetAllUserByCustomProperty (see page 426)	Returns all the users based on the given properties.
	GetAllUserGroups (see page 426)	Loads all of the user groups from the system. This method is overloaded.
	GetAllUsers (see page 427)	This method is overloaded.
	GetGroupsUserIsIn (see page 428)	Loads all the groups by user.
	GetMembershipUser (see page 428)	Returns user data based on the user's login name (username).
	GetUser (see page 428)	Loads the user details for the given user.
	GetUserByUsername (see page 429)	Loads the user details for the given user.
	GetUserCustomProperties (see page 429)	Returns an array of custom properties for a given user's ID.
	GetUserEmailInfo (see page 429)	Returns user data for a given user's ID.
	GetUserForFolderAdmin (see page 430)	
	GetUserGroup (see page 430)	Loads a given group's details.
	GetUserGroupByName (see page 430)	Loads a given group's details.
	GetUserGroupForFolderAdmin (see page 431)	Returns all the group's data for the administrator.

	GetUserPermissions (see page 431)	Loads the user permissions as a array of UserPermissionData.
	GetUserPreference (see page 431)	Loads the preferences for the given user.
	GetUsers (see page 432)	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.
	GetUsersForTask (see page 433)	Loads the users that have permissions to the given content (see page 337).
	GetUsersNotInGroup (see page 433)	Loads all the users that are not in the given group.
	IsAdmin (see page 434)	Determines the if the current user is an Administrator.
	IsARoleMember (see page 434)	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.
	IsARoleMemberForContent (see page 436)	Determines whether a Role Member is allowed to work with a given piece of content (see page 337) as an administrator or not. Administrations privileges are for the given piece of content (see page 337) 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 437)	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 437)	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 438)	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 438)	Determines whether a Role Member is allowed to work with a given folder as an meta data administrator or not.
	LogInUser (see page 438)	Used to log in the user.
	ReadCustomProperties (see page 439)	Returns the custom properties as a Visual Basic collection. Importing the Microsoft.VisualBasic name space might be required.
	ResetMembershipUserPassword (see page 439)	Reset the password for the given membership user.
	SetAuthenticationCookie (see page 439)	This method sets the authentication cookie.
	UnsubscribeUser (see page 440)	Unsubscribes a user from the system.
	UpdateCustomPropertiesItemOrder (see page 440)	Updates the order of available custom properties.
	UpdateUser (see page 441)	
	UpDateUserGroup (see page 441)	
	UpdateUserPreferences (see page 442)	Updates the given user's preference data.
	UserObject (see page 442)	Returns user data based on the user's ID or by the user's login or username. This function is overloaded.

Legend

	Method

ActivateUserAccount Method

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 (Integer) (see page 419)	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 419)	Activates the deleted (deactivated) user into the current/active state using the Username and GUID (AccountID) parameter. This function is overloaded. See Also: ActivateUserAccount (Integer)

User.ActivateUserAccount Method (Integer)

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(int UserId);
```

Visual Basic

```
Public Function ActivateUserAccount(ByVal UserId As Integer) As Boolean
```

Parameters

Parameters	Description
UserId	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 (Integer)

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
Username	The login name for the user.
AccountID	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
udUserData	An object of class which represents the corresponding user elements.

Returns

Ektron.Cms.UserData

User.AddNewUser Method

Adds a new 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
udUserData	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
RoleName	Required role name.

Returns

True if successfully added. Otherwise, returns false

AddUser Method

Overload List

	Name	Description
	User.AddUser (Collection) (see page 421)	Adds a new user to CMS.
	User.AddUser (UserData) (see page 421)	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
Users	User 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
Users	User details as a value, key pair.

Returns

Ektron.Cms.UserData

User.AddUserGroup Method

Creates new user group.

C#

```
public Collection AddUserGroup(String GroupName, String GroupPath, 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
GroupName	The name of the group.
GroupPath	Group path, usually required for Active Directory, otherwise CMS group assumed.
GroupDomain	Group domain, usually required for Active Directory.

User.AddUserToGroup Method

Adds an existing user to an existing group.

C#

```
public AddUserToGroup(int UserId, int GroupId);
```

Visual Basic

```
Public Sub AddUserToGroup(ByVal UserId As Integer, ByVal GroupId As Integer)
```

Parameters

Parameters	Description
UserId	The user's ID.
GroupId	The required group's ID.

User.AutoLogInUser Method

Login 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
SAMusername	user name
domain	domain
ServerName	server name

Returns

True if the user successfully logged in otherwise false.

CreateBlankUserObject Method

This method is overloaded.

Overload List

	Name	Description
	User.CreateBlankUserObject () (see page 422)	Returns the instance of the user data. This function is overloaded. See Also: CreateBlankUserObject (UserData)
	User.CreateBlankUserObject (UserData) (see page 423)	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

C#

```
public CreateBlankUserObject(UserData udUserData);
```

Visual Basic

```
Public Sub CreateBlankUserObject(ByRef udUserData As UserData)
```

Parameters

Parameters	Description
udUserData	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.

C#

```
public DeleteUserByID(int Id);
```

Visual Basic

```
Public Sub DeleteUserByID(ByVal Id As Integer)
```

Parameters

Parameters	Description
Id	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
Ids	The selected user IDs (comma separated values).

User.DeleteUserFromGroup Method

Removes a user from the designated group.

C#

```
public DeleteUserFromGroup(int UserId, int GroupId, Boolean ReturnDeleted);
```

Visual Basic

```
Public Sub DeleteUserFromGroup(ByVal UserId As Integer, ByVal GroupId As Integer, Optional
ByVal ReturnDeleted As Boolean = False)
```

Parameters

Parameters	Description
UserId	The user's ID.

GroupId	The group's ID.

User.DeleteUserGroup Method

Removes the group from the system permanently.

C#

```
public DeleteUserGroup(int Id);
```

Visual Basic

```
Public Sub DeleteUserGroup(ByVal Id As Integer)
```

Parameters

Parameters	Description
Id	The group's ID.

User.EditUserCustomProperties Method

This method creates the editable fields(UI) for user's custom properties.

C#

```
public String EditUserCustomProperties(int UserId, Boolean bReadOnly);
```

Visual Basic

```
Public Function EditUserCustomProperties(Optional ByVal UserId As Integer = 0, Optional ByVal
bReadOnly As Boolean = False) As String
```

Parameters

Parameters	Description
UserId	User's ID. If the user ID is not specified then it will return empty fields.
bReadOnly	

Returns

String

User.GetActiveUser Method

Returns all of the user properties with or without preference.

C#

```
public UserData GetActiveUser(int Id, Boolean Preference);
```

Visual Basic

```
Public Function GetActiveUser(ByVal Id As Integer, Optional ByVal Preference As Boolean =
False) As UserData
```

Parameters

Parameters	Description
Id	The user's ID.
Preference	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 name.

C#

```
public AdMappingData GetADMapping(int CallerId, String MapType, int Map, int Exclude, int
Editable);
```

Visual Basic

```
Public Function GetADMapping(ByVal CallerId As Integer, ByVal MapType As String, ByVal Map As
Integer, ByVal Exclude As Integer, ByVal Editable As Integer) As AdMappingData
```

Parameters

Parameters	Description
CallerId	Who is calling this function.
MapType	Possible value is “userprop” (user properties).
Map	Ektron.Cms.Common.EkEnumeration.MapTypes (include, exclude and dontconsider are the possible values).
Exclude	Flag (0 – false, 1-true).
Editable	Flag(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
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

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
PropType	
PropValues	Required property

GetAllUserGroups Method

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 426)	Loads all of the user groups from the system. This function is overloaded. See Also: GetAllUserGroups (String)
	User.GetAllUserGroups (String) (see page 427)	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
GroupType	GroupType: 1 = Membership (see page 212); 0 = CMSUser

OrderBy	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
OrderBy	Arranges data in a specific order. Possible values are **GroupName** or **GroupID**.

Returns

Ektron.Cms.UserGroupData()

GetAllUsers Method

This method is overloaded.

Overload List

	Name	Description
	User.GetAllUsers (Integer, Integer) (see page 427)	Gets the User Group Data of all the allowed users in the folder specified.
	User.GetAllUsers (UserRequestData) (see page 428)	Returns the list of users from the database based on the UserRequestData class.

User.GetAllUsers Method (Integer, Integer)

Gets the User Group Data of all the allowed users in the folder specified.

C#

```
public UserGroupData GetAllUsers(int CallerID, int FolderId);
```

Visual Basic

```
Public Function GetAllUsers(ByVal CallerID As Integer, ByVal FolderId As Integer) As
UserGroupData
```

Parameters

Parameters	Description
CallerID	The user ID that is logged on to the system.
FolderId	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(UserRequestData userRequest);
```

Visual Basic

```
Public Function GetAllUsers(ByRef userRequest As UserRequestData) As UserData
```

Parameters

Parameters	Description
userRequest	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(int UserId, String OrderBy);
```

Visual Basic

```
Public Function GetGroupsUserIsIn(ByVal UserId As Integer, ByVal OrderBy As String) As
GroupData
```

Parameters

Parameters	Description
UserId	The user's ID.
OrderBy	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
Username	User's login name (username). This is required

Returns

Ektron.Cms.UserData

User.GetUser Method

Loads the user details for the given user.

C#

```
public UserData GetUser(int Id, Boolean Preference, Boolean ReturnDeleted);
```

Visual Basic

```
Public Function GetUser(ByVal Id As Integer, Optional ByVal Preference As Boolean = False,
Optional ByVal ReturnDeleted As Boolean = False) As UserData
```

Parameters

Parameters	Description
Id	The user ID.
Preference	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
username	The username.

Returns

Ektron.Cms.UserData

User.GetUserCustomProperties Method

Returns an array of custom properties for a given user's ID.

C#

```
public CustomAttribute GetUserCustomProperties(int UserId);
```

Visual Basic

```
Public Function GetUserCustomProperties(ByVal UserId As Integer) As CustomAttribute
```

Parameters

Parameters	Description
UserId	User's ID

Returns

Ektron.Cms.CustomAttribute()

User.GetUserEmailInfo Method

Returns user data for a given user's ID.

C#

```
public UserData GetUserEmailInfo(int Id);
```

Visual Basic

```
Public Function GetUserEmailInfo(ByVal Id As Integer) As UserData
```

Parameters

Parameters	Description
Id	The user's ID.

Returns

Object of UserData

User.GetUserForFolderAdmin Method

C#

```
public UserData GetUserForFolderAdmin(int FolderId, int Id, Boolean Preference, Boolean
ReturnDeleted);
```

Visual Basic

```
Public Function GetUserForFolderAdmin(ByVal FolderId As Integer, ByVal Id As Integer, Optional
ByVal Preference As Boolean = False, Optional ByVal ReturnDeleted As Boolean = False) As
UserData
```

Parameters

Parameters	Description
FolderId	
Id	
Preference	

User.GetUserGroup Method

Loads a given group's details.

C#

```
public UserGroupData GetUserGroup(int Id);
```

Visual Basic

```
Public Function GetUserGroup(ByVal Id As Integer) As UserGroupData
```

Parameters

Parameters	Description
Id	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
name	The group's name.

Returns

Ektron.Cms.UserGroupData

User.GetUserGroupForFolderAdmin Method

Returns all the group's data for the administrator.

C#

```
public UserGroupData GetUserGroupForFolderAdmin(int FolderId, int Id);
```

Visual Basic

```
Public Function GetUserGroupForFolderAdmin(ByVal FolderId As Integer, ByVal Id As Integer) As
UserGroupData
```

Parameters

Parameters	Description
FolderId	Folder ID
Id	User ID

Returns

UserGroupData

User.GetUserPermissions Method

Loads the user permissions as a array of UserPermissionData.

C#

```
public UserPermissionData GetUserPermissions(int Id, String ItemType, int UserId, String
UserGroupList, PermissionUserType PermissionType, PermissionRequestType PermissionRequest);
```

Visual Basic

```
Public Function GetUserPermissions(ByVal Id As Integer, ByVal ItemType As String, ByVal UserId
As Integer, ByVal UserGroupList As String, Optional ByVal PermissionType As PermissionUserType
= PermissionUserType.All, Optional ByVal PermissionRequest As PermissionRequestType =
PermissionRequestType.All) As UserPermissionData
```

Parameters

Parameters	Description
Id	The item's ID.
ItemType	The item's type.
UserId	The user's ID.
UserGroupList	The group list.
PermissionType	PermissionUserType
PermissionRequest	PermissionRequestType

Returns

Ektron.Cms.UserPermissionData()

User.GetUserPreference Method

Loads the preferences for the given user.

C#

```
public UserPreferenceData GetUserPreference(int Id);
```

Visual Basic

```
Public Function GetUserPreference(ByVal Id As Integer) As UserPreferenceData
```

Parameters

Parameters	Description
Id	The user's ID.

Returns

Ektron.Cms.UserPreferenceData

GetUsers Method

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 (Integer, String) (see page 432)	Loads the all the users from the given group.
	User.GetUsers (String, String) (see page 433)	Returns an array of user group data based on the userName and userNameType.

User.GetUsers Method (Integer, String)

Loads the all the users from the given group.

C#

```
public UserData GetUsers(int GroupId, String OrderBy);
```

Visual Basic

```
Public Function GetUsers(ByVal GroupId As Integer, ByVal OrderBy As String) As UserData
```

Parameters

Parameters	Description
GroupId	The group's ID
OrderBy	The order of the users. The string value can be: • username • firstname&amp • 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, String userNameTypes);
```

Visual Basic

```
Public Function GetUsers(ByVal userNames As String, Optional ByVal userNameTypes As String =
"") As UserGroupData
```

Parameters

Parameters	Description
userNames	The user's name. (Required)
userNameTypes	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 (see page 337).

C#

```
public UserGroupData GetUsersForTask(int CallerID, int ContentID);
```

Visual Basic

```
Public Function GetUsersForTask(ByVal CallerID As Integer, ByVal ContentID As Integer) As
UserGroupData
```

Parameters

Parameters	Description
CallerID	The user ID that is logged on to the system.
ContentID	The ID of the content (see page 337). (Required)

Returns

Ektron.Cms.UserGroupData()

User.GetUsersNotInGroup Method

Loads all the users that are not in the given group.

C#

```
public UserData GetUsersNotInGroup(int Id, String OrderBy, int GroupType);
```

Visual Basic

```
Public Function GetUsersNotInGroup(ByVal Id As Integer, ByVal OrderBy As String, Optional
ByVal GroupType As Integer = 0) As UserData
```

Parameters

Parameters	Description
Id	The group ID.

`OrderBy`	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.

IsARoleMember Method

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 435)	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) and CommonApi.IsARoleMember Method (Integer).
	User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean) (see page 435)	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) and CommonApi.IsARoleMember Method (Integer).

	User.IsARoleMember (Integer) (see page 436)	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) and CommonApi.IsARoleMember Method (Integer, Integer, Boolean).
	User.IsARoleMember (Integer, Integer, Boolean) (see page 436)	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) and CommonApi.IsARoleMember Method (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: CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean), CommonApi.IsARoleMember Method (Integer, Integer, Boolean) and CommonApi.IsARoleMember Method (Integer).

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
RoleId	

User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, 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: CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds), CommonApi.IsARoleMember Method (Integer, Integer, Boolean) and CommonApi.IsARoleMember Method (Integer).

C#

```
public Boolean IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId, int nUserId,
Boolean GroupFlag);
```

Visual Basic

```
Public Function IsARoleMember(ByVal RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds,
ByVal nUserId As Integer, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
RoleId	
nUserId	
GroupFlag	

User.IsARoleMember Method (Integer)

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) and CommonApi.IsARoleMember Method (Integer, Integer, Boolean).

C#

```
public Boolean IsARoleMember(int RoleId);
```

Visual Basic

```
Public Function IsARoleMember(ByVal RoleId As Integer) As Boolean
```

Parameters

Parameters	Description
RoleId	

User.IsARoleMember Method (Integer, Integer, 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: CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean), CommonApi.IsARoleMember Method (Integer) and CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds).

C#

```
public Boolean IsARoleMember(int RoleId, int nUserId, Boolean GroupFlag);
```

Visual Basic

```
Public Function IsARoleMember(ByVal RoleId As Integer, ByVal nUserId As Integer, Optional
ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
RoleId	
nUserId	
GroupFlag	

User.IsARoleMemberForContent Method

Determines whether a Role Member is allowed to work with a given piece of content (see page 337) as an administrator or not. Administrations privileges are for the given piece of content (see page 337) 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(int RoleId, int ContentId, int nUserId, Boolean
```

```
GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForContent(ByVal RoleId As Integer, ByVal ContentId As Integer,
Optional ByVal nUserId As Integer = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
RoleId	
ContentId	The numeric content (see page 337) ID for which you are giving the Role Member privileges.
nUserId	
GroupFlag	

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(int ContentId, int nUserId, Boolean
GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForContent_FolderUserAdmin(ByVal ContentId As Integer, Optional
ByVal nUserId As Integer = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
ContentId	required content (see page 337) id
nUserId	user id
GroupFlag	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(int RoleId, int FolderId, int nUserId, Boolean
GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForFolder(ByVal RoleId As Integer, ByVal FolderId As Integer,
Optional ByVal nUserId As Integer = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
RoleId	
FolderId	The numeric folder ID for which you are giving the Role Member privileges.
nUserId	
GroupFlag	

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(int FolderId, int nUserId, Boolean
GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForFolder_FolderUserAdmin(ByVal FolderId As Integer, Optional
ByVal nUserId As Integer = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
FolderId	required folder id
nUserId	user id
GroupFlag	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(int FolderId, int nUserId, Boolean
GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForFolder_MetadataAdmin(ByVal FolderId As Integer, Optional ByVal
nUserId As Integer = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
FolderId	required folder id
nUserId	user id
GroupFlag	group id

Returns

True if granted otherwise false

User.LogInUser Method

Used to log in the user.

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, Optional ByVal Domain As String = "", Optional ByVal Protocol As String = "") As
UserData
```

Parameters

Parameters	Description
username	The system username.
Password	The system password.
ServerName	The system server name.
Domain	The user's domain, required only when users login using Active Directory.
Protocol	Required only when using a secure login or when you are using any valid Active Directory protocols.

Returns

Ektron.Cms.UserData

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
form	A collection of user properties.

Returns

Microsoft.VisualBasic.Collection (see page 63)

User.ResetMembershipUserPassword Method

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
Username	User's username to be reset.

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
userInfo	Returned value of the logInUser (see page 438) method.

Remarks

First, you must call the logInUser (see page 438) method to login the user.

User.UnsubscribeUser Method

Unsubscribes a user from the system.

C#

```
public Boolean UnsubscribeUser(String Username, String Password);
```

Visual Basic

```
Public Function UnsubscribeUser(ByVal Username As String, Optional ByVal Password As String =
"") As Boolean
```

Parameters

Parameters	Description
Username	The user's username.
Password	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
ItemOrder	Order of the properties applicable to the user.

Returns

Boolean

UpdateUser Method

Overload List

	Name	Description
	User.UpdateUser (Collection) (see page 441)	Updates the user details.
	User.UpdateUser (UserData) (see page 441)	Updates the user details.

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
data	A collection of user data.

User.UpdateUser Method (UserData)

Updates the user details.

C#

```
public UpdateUser(UserData data);
```

Visual Basic

```
Public Sub UpdateUser(ByVal data As UserData)
```

Parameters

Parameters	Description
data	A UserData object of user data.

UpDateUserGroup Method

Overload List

	Name	Description
	User.UpDateUserGroup (Collection) (see page 441)	Updates the user group details.
	User.UpDateUserGroup (UserData) (see page 442)	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
Group	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
data	data As UserData)

User.UpdateUserPreferences Method

Updates the given user's preference data.

C#

```
public UpdateUserPreferences(int Id, Collection data);
```

Visual Basic

```
Public Sub UpdateUserPreferences(ByVal Id As Integer, ByVal data As Collection)
```

Parameters

Parameters	Description
Id	The user's ID.
data	A collection of user preferences.

UserObject Method

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 (Integer) (see page 442)	Returns user data based on the user's ID. This function is overloaded. See Also: UserObject (String)
	User.UserObject (String) (see page 443)	Returns user data based on the user's login or username. This function is overloaded. See Also: UserObject (Integer)

User.UserObject Method (Integer)

Returns user data based on the user's ID.

This function is overloaded. See Also: UserObject (String)

C#

```
public UserData UserObject(int Id);
```

Visual Basic

```
Public Function UserObject(ByVal Id As Integer) As UserData
```

Parameters

Parameters	Description
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 (Integer)

C#

```
public UserData UserObject(String sUserName);
```

Visual Basic

```
Public Function UserObject(ByVal sUserName As String) As UserData
```

Parameters

Parameters	Description
sUserName	User's login or username.

Returns

Ektron.Cms.UserData

Classes

The following table lists classes in this documentation.

Classes

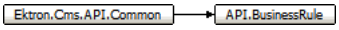
	Name	Description
	BusinessRule (see page 444)	This is class API.BusinessRule.
	CustomFields (see page 446)	This is class API.CustomFields.
	DateTimeSelector (see page 458)	This is class API.DateTimeSelector.
	Folder (see page 466)	This is class API.Folder.
	Font (see page 472)	This is class API.Font.
	Library (see page 475)	This is class API.Library.
	Localization (see page 484)	This is class API.Localization.
	Metadata (see page 495)	This is class API.Metadata.
	Permissions (see page 500)	This is class API.Permissions.
	Site (see page 505)	This is class API.Site.
	SiteMap (see page 512)	This is class API.SiteMap.
	Subscription (see page 515)	This is class API.Subscription.
	Utilities (see page 524)	This is class API.Utilities.

Legend

	Class

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

This is class API.BusinessRule.

Members

BusinessRule Methods

	Name	Description
	GetBusinessRules (see page 445)	Method to get the results from a business rule. This method is overloaded.

Topics

Name	Description
BusinessRule Methods (see page 444)	The methods of the BusinessRule class are listed here.

Legend

	Method

BusinessRule Methods

	Name	Description
	GetBusinessRules (see page 445)	Method to get the results from a business rule. This method is overloaded.

Legend

	Method

BusinessRule Methods

The methods of the BusinessRule class are listed here.

Public Methods

	Name	Description
	GetBusinessRules (see page 445)	Method to get the results from a business rule. This method is overloaded.

Legend

	Method

GetBusinessRules Method

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, Integer) (see page 445)	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 445)	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, Integer)

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, int 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 Integer) As String
```

Parameters

Parameters	Description
`httpHost`	The HTTP host the request is made on.
`urlRest`	not used
`cookies`	Cookie Value Collection (see page 63)
`formValues`	Form Value Collection (see page 63)
`queryValues`	Querystring Value Collection (see page 63)
`serverValues`	Server Variables Collection (see page 63)
`paramValues`	Custom Value Collection (see page 63)
`iRuleID`	Integer 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

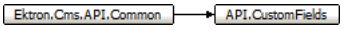
Parameters	Description
httpHost	The HTTP host the request is made on.
urlRest	not used
cookies	Cookie Value Collection (see page 63)
formValues	Form Value Collection (see page 63)
queryValues	Querystring Value Collection (see page 63)
serverValues	Server Variables Collection (see page 63)
paramValues	Custom Value Collection (see page 63)
sRuleID	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 API.CustomFields.

Members

CustomFields Methods

	Name	Description
	AddNewEditableCustomFieldAssignments (see page 450)	Returns the HTML string for the add/edit custom fields.
S	BoolToYesNo (see page 450)	Returns labels for yes or no based on locale.

	GetEditableCustomFieldAssignments (see page 450)	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 450)
S	GetEnhancedMetadataArea (see page 451)	Returns the HTML needed to render the EnhancedMetadata selection area.
S	GetEnhancedMetadataScript (see page 451)	Returns support javascript code for enhanced metadata.
	GetFieldsByFolder (see page 451)	Returns all of the custom attributes by folder.
S	GetSearchProperties (see page 452)	Returns the search's properties. This method is overloaded.
S	Populate_AssetRequestObjectFromForm (see page 453)	Returns asset request object using form data.
	ProcessCustomFields (see page 454)	Process Folders' Custom-Field (and metadata) assignments: When form submitted, copy checked items to hidden field. Form 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 454)	Returns the search result based on assets.
S	WriteFilteredMetadataForEdit (see page 454)	Returns an HTML string for filtered metadata.
S	WriteFilteredMetadataForView (see page 455)	Returns an HTML string for metadata view.
S	WriteLibrarySearchExtended (see page 455)	' Returns an HTML string for the library search custom metadata properties.
S	WriteMetadataDefaultForEdit (see page 455)	Returns an HTML string for the default metadata.
S	WriteMetadataTypeForView (see page 456)	Returns a metadata type for view.
S	WritePublicSearch (see page 456)	Returns WritepublicsearchEx (see page 457).
S	WritePublicSearchEx (see page 457)	Returns an HTML string for extended the search attributes.

Topics

Name	Description
CustomFields Methods (see page 449)	The methods of the CustomFields class are listed here.

Legend

	Method
S	static

CustomFields Methods

	Name	Description
	AddNewEditableCustomFieldAssignments (see page 450)	Returns the HTML string for the add/edit custom fields.
S	BoolToYesNo (see page 450)	Returns labels for yes or no based on locale.
	GetEditableCustomFieldAssignments (see page 450)	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 450)
S	GetEnhancedMetadataArea (see page 451)	Returns the HTML needed to render the EnhancedMetadata selection area.
S	GetEnhancedMetadataScript (see page 451)	Returns support javascript code for enhanced metadata.
	GetFieldsByFolder (see page 451)	Returns all of the custom attributes by folder.
S	GetSearchProperties (see page 452)	Returns the search's properties. This method is overloaded.
S	Populate_AssetRequestObjectFromForm (see page 453)	Returns asset request object using form data.
	ProcessCustomFields (see page 454)	Process Folders' Custom-Field (and metadata) assignments: When form submitted, copy checked items to hidden field. Form 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 454)	Returns the search result based on assets.
S	WriteFilteredMetadataForEdit (see page 454)	Returns an HTML string for filtered metadata.
S	WriteFilteredMetadataForView (see page 455)	Returns an HTML string for metadata view.
S	WriteLibrarySearchExtended (see page 455)	' Returns an HTML string for the library search custom metadata properties.
S	WriteMetadataDefaultForEdit (see page 455)	Returns an HTML string for the default metadata.
S	WriteMetadataTypeForView (see page 456)	Returns a metadata type for view.
S	WritePublicSearch (see page 456)	Returns WritepublicsearchEx (see page 457).
S	WritePublicSearchEx (see page 457)	Returns an HTML string for extended the search attributes.

Legend

	Method
S	static

CustomFields Methods

The methods of the CustomFields class are listed here.

Public Methods

	Name	Description
	AddNewEditableCustomFieldAssignments (see page 450)	Returns the HTML string for the add/edit custom fields.
S	BoolToYesNo (see page 450)	Returns labels for yes or no based on locale.
	GetEditableCustomFieldAssignments (see page 450)	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 450)
S	GetEnhancedMetadataArea (see page 451)	Returns the HTML needed to render the EnhancedMetadata selection area.
S	GetEnhancedMetadataScript (see page 451)	Returns support javascript code for enhanced metadata.
	GetFieldsByFolder (see page 451)	Returns all of the custom attributes by folder.
S	GetSearchProperties (see page 452)	Returns the search's properties. This method is overloaded.
S	Populate_AssetRequestObjectFromForm (see page 453)	Returns asset request object using form data.
	ProcessCustomFields (see page 454)	Process Folders' Custom-Field (and metadata) assignments: When form submitted, copy checked items to hidden field. Form 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 454)	Returns the search result based on assets.
S	WriteFilteredMetadataForEdit (see page 454)	Returns an HTML string for filtered metadata.
S	WriteFilteredMetadataForView (see page 455)	Returns an HTML string for metadata view.
S	WriteLibrarySearchExtended (see page 455)	' Returns an HTML string for the library search custom metadata properties.
S	WriteMetadataDefaultForEdit (see page 455)	Returns an HTML string for the default metadata.
S	WriteMetadataTypeForView (see page 456)	Returns a metadata type for view.
S	WritePublicSearch (see page 456)	Returns WritepublicsearchEx (see page 457).
S	WritePublicSearchEx (see page 457)	Returns an HTML string for extended the search attributes.

Legend

	Method
	static

CustomFields.AddNewEditableCustomFieldAssignments Method

Returns the HTML string for the add/edit custom fields.

C#

```
public String AddNewEditableCustomFieldAssignments(int parentfolderid);
```

Visual Basic

```
Public Function AddNewEditableCustomFieldAssignments(ByVal parentfolderid As Integer) As String
```

Parameters

Parameters	Description
parentfolderid	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
bValue	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 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(folderID, editMode);
```

Visual Basic

```
Public Function GetEditableCustomFieldAssignments(ByVal  As folderID, ByVal  As editMode) As String
```

Parameters

Parameters	Description
folderID	The folder's ID.
editMode	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, Long LanguageID);
```

Visual Basic

```
Public Function GetFieldsByFolder(ByVal FolderID As Long, ByVal LanguageID As Long) As
Collection
```

Parameters

Parameters	Description
FolderID	The folder's ID.
LanguageID	The language's ID.

Returns

Microsoft.VisualBasic.Collection (see page 63)

GetSearchProperties Method

Returns the search's properties. This method is overloaded.

Overload List

	Name	Description
	CustomFields.GetSearchProperties (Ektron.Cms.Common.SearchAssetDispayRequest) (see page 452)	Returns an HTML string for the search properties display.
	CustomFields.GetSearchProperties (String, String, String, String, Collection, String, String, String) (see page 452)	Returns an HTML string to display search properties.
	CustomFields.GetSearchProperties (System.Collections.Specialized.NameValueCollection, Ektron.Cms.Common.SearchItemCustom,), Ektron.Cms.Common.SearchItemCustom,)) (see page 453)	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
sadReq	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 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)
```

Parameters

Parameters	Description
MonikerStyle	Style. (Required)
DataType	The data type such as String, DateTime, Numeric, etc.
Caption	The caption value. (Required)
Name	The value for name attributes.
Items	Values of items.
AlignCaption	The value for the align attributes.
szBeginFontInfo	The value for the begin font attributes.
szEndFontInfo	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,
Ektron.Cms.Common.SearchItemCustom CustomProps, ), 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,  As ), ByRef AssetProps As
Ektron.Cms.Common.SearchItemCustom,  As ))
```

Parameters

Parameters	Description
FormReq	Form field values.
CustomProps	Custom properties value.
AssetProps	Asset properties value.

CustomFields.Populate_AssetRequestObjectFromForm Method

Returns asset request object using form data.

C#

```
public Shared Ektron.Cms.Common.SearchAssetRequest
Populate_AssetRequestObjectFromForm(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 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 ProcessCustomFields(Long folderID);
```

Visual Basic

```
Public Function ProcessCustomFields(ByVal folderID As Long)
```

Parameters

Parameters	Description
folderID	The folder's ID.

Returns

Object

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
sadReq	Search 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, int Count, 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
cMetadataTypes	Metadata (see page 495) types.
bSearchableOnly	bSearchableOnly
ty	type
FolderId	The folder's ID.
Count	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
cMetadataTypes	Metadata (see page 495) types.
FolderId	The folder's Id.
bSearchableOnly	True or False

Returns

String

CustomFields.WriteLibrarySearchExtended Method

C#

```
public Shared System.Text.StringBuilder WriteLibrarySearchExtended(int folderId);
```

Visual Basic

```
Public Shared Function WriteLibrarySearchExtended(ByVal folderId As Integer) As
System.Text.StringBuilder
```

Parameters

Parameters	Description
folderId	The folder's ID.

Returns

System.Text.StringBuilder

Description

' Returns an HTML string for the library search custom metadata properties.

CustomFields.WriteMetadataDefaultForEdit Method

Returns an HTML string for the default metadata.

C#

```
public Shared System.Text.StringBuilder WriteMetadataDefaultForEdit(Ektron.Cms.ContentMetaData
cMetadataType, int DataType, 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
cMetadataType	Metadata (see page 495) values.
DataType	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
DataType	Data types.
Caption	Caption value.
Value	Value. (Required)

Returns

String

CustomFields.WritePublicSearch Method

Returns WritepublicsearchEx (see page 457).

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
sadReq	Search 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, Boolean
EnableContentFieldset, String ContentFieldsetLegend, Boolean EnableLibraryFieldset, String
LibraryFieldsetLegend, Boolean EnableFolderFieldset, String FolderFieldsetLegend, Boolean
EnableCallBack, Boolean RemoveTeaserHtml, 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

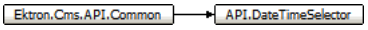
Parameters	Description
sadReq	Search asset properties. (Required)
EnableAdvancedLink	True or False
EnableContentBtn	True or False
EnableContent	True or False
EnableFormsBtn	True or False
EnableForms	True or False
EnableDMSAssetsBtn	True or False
EnableDMSAssets	True or False
EnableLibImagesBtn	True or False
EnableLibImages	True or False
EnableLibFilesBtn	True or False
EnableLibFiles	True or False
EnableLibHyperlinksBtn	True or False
EnableLibHyperlinks	True or False
EnableArchivedBtn	True or False
EnableArchived	True or False
TargetFolderDescriptionList	True or False
EnableContentFieldset	True or False
ContentFieldsetLegend	True or False
EnableLibraryFieldset	True or False
LibraryFieldsetLegend	True or False
EnableFolderFieldset	True or False
FolderFieldsetLegend	True or False

Returns

System.Text.StringBuilder

DateTimeSelector Class

Class Hierarchy



C#

```
public class DateTimeSelector : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class DateTimeSelector
Inherits Ektron.Cms.API.Common
```

File

DateTimeSelector.vb

Description

This is class API.DateTimeSelector.

Members

DateTimeSelector Methods

	Name	Description
	DisplayCultureDate (see page 460)	Returns server culture date using culture object.
	DisplayCultureDateTime (see page 460)	Returns server culture date and time.
	DisplayCultureTime (see page 461)	Returns the server culture time.
	RoundMinutes (see page 461)	Math function to round the minute.

Topics

Name	Description
DateTimeSelector Methods (see page 460)	The methods of the DateTimeSelector class are listed here.
DateTimeSelector Properties (see page 461)	The properties of the DateTimeSelector class are listed here.

DateTimeSelector Properties

	Name	Description
R	ClientDateSeparator (see page 462)	Returns client date separator value.
R	ClientLongDatePattern (see page 462)	Returns the client date in long date format.
R	ClientLongTimePattern (see page 463)	Returns the clients time in long format.
R	ClientShortDatePattern (see page 463)	Returns the clients date in short format.
R	ClientShortTimePattern (see page 463)	Returns the client time in short format.
R	ClientTimeSeparator (see page 463)	Returns the clients time separator.
	ExtendedMeta (see page 464)	Returns the value of meta extended or not meta extended.
	FormElement (see page 464)	Sets or gets the form element value.
	FormName (see page 464)	Sets or gets the form name.
	RequestInfo (see page 464)	Gets or sets the RequestInfo object.
R	ServerDateSeparator (see page 465)	Returns the server side date separator.

	ServerShortDatePattern (see page 465)	Returns the server side short date pattern.
	ServerShortTimePattern (see page 465)	Returns the server short time pattern.
	ServerTimeSeparator (see page 465)	Returns the server side date separator.
	SpanAttributes (see page 465)	Gets or sets the span attributes.
	SpanId (see page 466)	Gets or sets span for calendar (see page 336) control.
	TargetDate (see page 466)	Sets or gets the target date.
	ValidationType (see page 466)	Gets or sets the validation type.

Legend

	Method
	Property
	read only

DateTimeSelector Methods

	Name	Description
	DisplayCultureDate (see page 460)	Returns server culture date using culture object.
	DisplayCultureDateTime (see page 460)	Returns server culture date and time.
	DisplayCultureTime (see page 461)	Returns the server culture time.
	RoundMinutes (see page 461)	Math function to round the minute.

Legend

	Method

DateTimeSelector Properties

	Name	Description
	ClientDateSeparator (see page 462)	Returns client date separator value.
	ClientLongDatePattern (see page 462)	Returns the client date in long date format.
	ClientLongTimePattern (see page 463)	Returns the clients time in long format.
	ClientShortDatePattern (see page 463)	Returns the clients date in short format.
	ClientShortTimePattern (see page 463)	Returns the client time in short format.
	ClientTimeSeparator (see page 463)	Returns the clients time separator.
	ExtendedMeta (see page 464)	Returns the value of meta extended or not meta extended.
	FormElement (see page 464)	Sets or gets the form element value.
	FormName (see page 464)	Sets or gets the form name.
	RequestInfo (see page 464)	Gets or sets the RequestInfo object.
	ServerDateSeparator (see page 465)	Returns the server side date separator.
	ServerShortDatePattern (see page 465)	Returns the server side short date pattern.
	ServerShortTimePattern (see page 465)	Returns the server short time pattern.
	ServerTimeSeparator (see page 465)	Returns the server side date separator.

	SpanAttributes (see page 465)	Gets or sets the span attributes.
	SpanId (see page 466)	Gets or sets span for calendar (see page 336) control.
	TargetDate (see page 466)	Sets or gets the target date.
	ValidationType (see page 466)	Gets or sets the validation type.

Legend

	Property
	read only

DateTimeSelector Methods

The methods of the DateTimeSelector class are listed here.

Public Methods

	Name	Description
	DisplayCultureDate (see page 460)	Returns server culture date using culture object.
	DisplayCultureDateTime (see page 460)	Returns server culture date and time.
	DisplayCultureTime (see page 461)	Returns the server culture time.
	RoundMinutes (see page 461)	Math function to round the minute.

Legend

	Method

DateTimeSelector.DisplayCultureDate Method

Returns server culture date using culture object.

C#

```
public String DisplayCultureDate(Boolean isEditable, String inSpanID, String inFormElemName);
```

Visual Basic

```
Public Function DisplayCultureDate(ByVal isEditable As Boolean, Optional ByVal inSpanID As
String = "", Optional ByVal inFormElemName As String = "") As String
```

Parameters

Parameters	Description
isEditable	True for editable span tag, otherwise false.
inSpanID	ID value for the span tag.
inFormElemName	Name value for the span tag.

Returns

String

DateTimeSelector.DisplayCultureDateTime Method

Returns server culture date and time.

C#

```
public String DisplayCultureDateTime(Boolean isEditable, String inSpanID, String
inFormElemName);
```

Visual Basic

```
Public Function DisplayCultureDateTime(ByVal isEditable As Boolean, Optional ByVal inSpanID As
String = "", Optional ByVal inFormElemName As String = "") As String
```

Parameters

Parameters	Description
isEditable	True for editable span tag, otherwise false.
inSpanID	Id value for the span tag.
inFormElemName	Name value for the span tag.

Returns

String

DateTimeSelector.DisplayCultureTime Method

Returns the server culture time.

C#

```
public String DisplayCultureTime(Boolean isEditable, String inSpanID, String inFormElemName);
```

Visual Basic

```
Public Function DisplayCultureTime(ByVal isEditable As Boolean, Optional ByVal inSpanID As
String = "", Optional ByVal inFormElemName As String = "") As String
```

Parameters

Parameters	Description
isEditable	True for editable span tag, otherwise false.
inSpanID	ID value for the span tag.
inFormElemName	Name value for the span tag.

DateTimeSelector.RoundMinutes Method

Math function to round the minute.

C#

```
public Date RoundMinutes(Date dateToRound, int minuteIncrement);
```

Visual Basic

```
Public Function RoundMinutes(ByVal dateToRound As Date, ByVal minuteIncrement As Integer) As
Date
```

Parameters

Parameters	Description
dateToRound	Date value
minuteIncrement	Minute value

Returns

Date

DateTimeSelector Properties

The properties of the DateTimeSelector class are listed here.

Public Properties

	Name	Description
R	ClientDateSeparator (see page 462)	Returns client date separator value.
R	ClientLongDatePattern (see page 462)	Returns the client date in long date format.
R	ClientLongTimePattern (see page 463)	Returns the clients time in long format.
R	ClientShortDatePattern (see page 463)	Returns the clients date in short format.
R	ClientShortTimePattern (see page 463)	Returns the client time in short format.
R	ClientTimeSeparator (see page 463)	Returns the clients time separator.
	ExtendedMeta (see page 464)	Returns the value of meta extended or not meta extended.
	FormElement (see page 464)	Sets or gets the form element value.
	FormName (see page 464)	Sets or gets the form name.
	RequestInfo (see page 464)	Gets or sets the RequestInfo object.
R	ServerDateSeparator (see page 465)	Returns the server side date separator.
R	ServerShortDatePattern (see page 465)	Returns the server side short date pattern.
R	ServerShortTimePattern (see page 465)	Returns the server short time pattern.
R	ServerTimeSeparator (see page 465)	Returns the server side date separator.
	SpanAttributes (see page 465)	Gets or sets the span attributes.
	SpanId (see page 466)	Gets or sets span for calendar (see page 336) control.
	TargetDate (see page 466)	Sets or gets the target date.
	ValidationType (see page 466)	Gets or sets the validation type.

Legend

	Property
R	read only

DateTimeSelector.ClientDateSeparator Property

Returns client date separator value.

C#

```
public ReadOnly String ClientDateSeparator;
```

Visual Basic

```
Public ReadOnly Property ClientDateSeparator() As String
```

Returns

String

DateTimeSelector.ClientLongDatePattern Property

Returns the client date in long date format.

C#

```
public ReadOnly String ClientLongDatePattern;
```

Visual Basic

```
Public ReadOnly Property ClientLongDatePattern() As String
```

Returns

String

DateTimeSelector.ClientLongTimePattern Property

Returns the clients time in long format.

C#

```
public ReadOnly String ClientLongTimePattern;
```

Visual Basic

```
Public ReadOnly Property ClientLongTimePattern() As String
```

Returns

String

DateTimeSelector.ClientShortDatePattern Property

Returns the clients date in short format.

C#

```
public ReadOnly String ClientShortDatePattern;
```

Visual Basic

```
Public ReadOnly Property ClientShortDatePattern() As String
```

Returns

String

DateTimeSelector.ClientShortTimePattern Property

Returns the client time in short format.

C#

```
public ReadOnly String ClientShortTimePattern;
```

Visual Basic

```
Public ReadOnly Property ClientShortTimePattern() As String
```

Returns

String

DateTimeSelector.ClientTimeSeparator Property

Returns the clients time separator.

C#

```
public ReadOnly String ClientTimeSeparator;
```

Visual Basic

```
Public ReadOnly Property ClientTimeSeparator() As String
```

Returns

String

DateTimeSelector.ExtendedMeta Property

Returns the value of meta extended or not meta extended.

C#

```
public Boolean ExtendedMeta;
```

Visual Basic

```
Public Property ExtendedMeta() As Boolean
```

Returns

Boolean

DateTimeSelector.FormElement Property

Sets or gets the form element value.

C#

```
public String FormElement;
```

Visual Basic

```
Public Property FormElement() As String
```

Returns

String

DateTimeSelector.FormName Property

Sets or gets the form name.

C#

```
public String FormName;
```

Visual Basic

```
Public Property FormName() As String
```

Returns

String

DateTimeSelector.RequestInfo Property

Gets or sets the RequestInfo object.

C#

```
public EkRequestInformation RequestInfo;
```

Visual Basic

```
Public Property RequestInfo() As EkRequestInformation
```

Returns

Ektron.Cms.Common.EkRequestInformation

DateTimeSelector.ServerDateSeparator Property

Returns the server side date separator.

C#

```
public ReadOnly String ServerDateSeparator;
```

Visual Basic

```
Public ReadOnly Property ServerDateSeparator() As String
```

Returns

String

DateTimeSelector.ServerShortDatePattern Property

Returns the server side short date pattern.

C#

```
public ReadOnly String ServerShortDatePattern;
```

Visual Basic

```
Public ReadOnly Property ServerShortDatePattern() As String
```

Returns

String

DateTimeSelector.ServerShortTimePattern Property

Returns the server short time pattern.

C#

```
public ReadOnly String ServerShortTimePattern;
```

Visual Basic

```
Public ReadOnly Property ServerShortTimePattern() As String
```

Returns

String

DateTimeSelector.ServerTimeSeparator Property

Returns the server side date separator.

C#

```
public ReadOnly String ServerTimeSeparator;
```

Visual Basic

```
Public ReadOnly Property ServerTimeSeparator() As String
```

Returns

String

DateTimeSelector.SpanAttributes Property

Gets or sets the span attributes.

C#

```
public String SpanAttributes;
```

Visual Basic

```
Public Property SpanAttributes() As String
```

Returns

String

DateTimeSelector.SpanId Property

Gets or sets span for calendar (see page 336) control.

C#

```
public String SpanId;
```

Visual Basic

```
Public Property SpanId() As String
```

Returns

String

DateTimeSelector.TargetDate Property

Sets or gets the target date.

C#

```
public Date TargetDate;
```

Visual Basic

```
Public Property TargetDate() As Date
```

Returns

Date

DateTimeSelector.ValidationType Property

Gets or sets the validation type.

C#

```
public int ValidationType;
```

Visual Basic

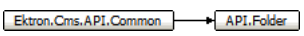
```
Public Property ValidationType() As Integer
```

Returns

Integer

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 API.Folder.

Members

Folder Methods

	Name	Description
	AddFolder (see page 468)	Creates CMS folder
	CreateFolder (see page 468)	Creates a folder.
	DeleteFolderById (see page 469)	Delete the folder
	GetChildFolders (see page 469)	Loads all the subfolders for a given folder ID.
	GetFolder (see page 469)	Retrieves the folder's details.
	GetFolderDataWithPermission (see page 470)	Retrieves the folder's details.
	GetFolderId (see page 470)	
	GetFolderInfo (see page 471)	Returns the folder detail.
	GetPath (see page 471)	Find's a parent ID by using a given folder ID.
	RenameFolder (see page 472)	rename the folder with new name
	UpdateFolder (see page 472)	Update CMS folder

Topics

Name	Description
Folder Methods (see page 468)	The methods of the Folder class are listed here.

Legend

	Method
	virtual

Folder Methods

	Name	Description
	AddFolder (see page 468)	Creates CMS folder
	CreateFolder (see page 468)	Creates a folder.
	DeleteFolderById (see page 469)	Delete the folder
	GetChildFolders (see page 469)	Loads all the subfolders for a given folder ID.
	GetFolder (see page 469)	Retrieves the folder's details.
	GetFolderDataWithPermission (see page 470)	Retrieves the folder's details.
	GetFolderId (see page 470)	
	GetFolderInfo (see page 471)	Returns the folder detail.
	GetPath (see page 471)	Find's a parent ID by using a given folder ID.
	RenameFolder (see page 472)	rename the folder with new name

	UpdateFolder (see page 472)	Update CMS folder

Legend

	Method
	virtual

Folder Methods

The methods of the Folder class are listed here.

Public Methods

	Name	Description
	AddFolder (see page 468)	Creates CMS folder
	CreateFolder (see page 468)	Creates a folder.
	DeleteFolderById (see page 469)	Delete the folder
	GetChildFolders (see page 469)	Loads all the subfolders for a given folder ID.
	GetFolder (see page 469)	Retrieves the folder's details.
	GetFolderDataWithPermission (see page 470)	Retrieves the folder's details.
	GetFolderId (see page 470)	
	GetFolderInfo (see page 471)	Returns the folder detail.
	GetPath (see page 471)	Find's a parent ID by using a given folder ID.
	RenameFolder (see page 472)	rename the folder with new name
	UpdateFolder (see page 472)	Update CMS folder

Legend

	Method
	virtual

Folder.AddFolder Method

Creates CMS folder

C#

```
public AddFolder(FolderRequest request);
```

Visual Basic

```
Public Sub AddFolder(ByVal request As FolderRequest)
```

Parameters

Parameters	Description
request	folder properties

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
Path	The path on the IIS server. (Required)

Returns

Boolean

Folder.DeleteFolderById Method

Delete the folder

C#

```
public Boolean DeleteFolderById(int Id);
```

Visual Basic

```
Public Function DeleteFolderById(ByVal Id As Integer) As Boolean
```

Parameters

Parameters	Description
Id	The folder id

Returns

False=if successfully deleted otherwise True

Folder.GetChildFolders Method

Loads all the subfolders for a given folder ID.

C#

```
public FolderData GetChildFolders(int Id, Boolean Recursive, String OrderBy);
```

Visual Basic

```
Public Function GetChildFolders(ByVal Id As Integer, Optional ByVal Recursive As Boolean =
False, Optional ByVal OrderBy As String = "name") As FolderData
```

Parameters

Parameters	Description
Id	The folder ID
Recursive	True = recursive look up. False = none
OrderBy	An order by clause

Returns

Ektron.Cms.FolderData()

Folder.GetFolder Method

Retrieves the folder's details.

C#

```
public FolderData GetFolder(int Id);
```

Visual Basic

```
Public Function GetFolder(ByVal Id As Integer) As FolderData
```

Parameters

Parameters	Description
Id	The folder's ID

Returns

Ektron.Cms.FolderData

Remarks

Read permission required

Folder.GetFolderDataWithPermission Method

Retrieves the folder's details.

C#

```
public FolderData GetFolderDataWithPermission(int Id);
```

Visual Basic

```
Public Function GetFolderDataWithPermission(ByVal Id As Integer) As FolderData
```

Parameters

Parameters	Description
Id	The folder's ID

Returns

Ektron.Cms.FolderData

GetFolderId Method

Overload List

	Name	Description
	Folder.GetFolderId (Integer) (see page 470)	Finds the folder ID by using the a content (see page 337)'s ID.
	Folder.GetFolderId (String, Integer) (see page 471)	Finds the folder ID by folder name and in a parent folder specified.

Folder.GetFolderId Method (Integer)

Finds the folder ID by using the a content (see page 337)'s ID.

C#

```
public int GetFolderId(int Id);
```

Visual Basic

```
Public Function GetFolderId(ByVal Id As Integer) As Integer
```

Parameters

Parameters	Description
Id	The content (see page 337)'s ID

Returns

Integer

Folder.GetFolderId Method (String, Integer)

Finds the folder ID by folder name and in a parent folder specified.

C#

```
public int GetFolderId(String FolderName, int ParentId);
```

Visual Basic

```
Public Function GetFolderId(ByVal FolderName As String, ByVal ParentId As Integer) As Integer
```

Parameters

Parameters	Description
FolderName	Name of the folder
ParentId	Look in a 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
Path	The folder's path

Returns

String()

Folder.GetPath Method

Find's a parent ID by using a given folder ID.

C#

```
public String GetPath(int Id);
```

Visual Basic

```
Public Function GetPath(ByVal Id As Integer) As String
```

Parameters

Parameters	Description
Id	The folder's ID

Returns

Integer

Folder.RenameFolder Method

rename the folder with new name

C#

```
public Boolean RenameFolder(int cId, String title);
```

Visual Basic

```
Public Function RenameFolder(ByVal cId As Integer, ByVal title As String) As Boolean
```

Parameters

Parameters	Description
cId	folder id
title	new folder name

Returns

False=if successfully deleted otherwise True

Folder.UpdateFolder Method

Update CMS folder

C#

```
public UpdateFolder(FolderRequest request);
```

Visual Basic

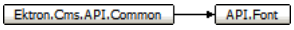
```
Public Sub UpdateFolder(ByVal request As FolderRequest)
```

Parameters

Parameters	Description
request	folder properties

Font Class

Class Hierarchy



C#

```
public class Font : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Font
Inherits Ektron.Cms.API.Common
```

File

Font.vb

Description

This is class API.Font.

Members

Font Methods

	Name	Description
	AddFont (see page 473)	Adds a new font to the system.

	DeleteFont (see page 474)	Deletes a font from the system.
	GetAllFonts (see page 474)	Loads all of the fonts as a array of FontData.
	GetFont (see page 474)	Loads the required font using the font's ID.
	GetFontConfigList (see page 474)	Returns all of the font details in a drop down control.
	UpdateFont (see page 475)	Updates the font.

Topics

Name	Description
Font Methods (see page 473)	The methods of the Font class are listed here.

Legend

	Method

Font Methods

	Name	Description
	AddFont (see page 473)	Adds a new font to the system.
	DeleteFont (see page 474)	Deletes a font from the system.
	GetAllFonts (see page 474)	Loads all of the fonts as a array of FontData.
	GetFont (see page 474)	Loads the required font using the font's ID.
	GetFontConfigList (see page 474)	Returns all of the font details in a drop down control.
	UpdateFont (see page 475)	Updates the font.

Legend

	Method

Font Methods

The methods of the Font class are listed here.

Public Methods

	Name	Description
	AddFont (see page 473)	Adds a new font to the system.
	DeleteFont (see page 474)	Deletes a font from the system.
	GetAllFonts (see page 474)	Loads all of the fonts as a array of FontData.
	GetFont (see page 474)	Loads the required font using the font's ID.
	GetFontConfigList (see page 474)	Returns all of the font details in a drop down control.
	UpdateFont (see page 475)	Updates the font.

Legend

	Method

Font.AddFont Method

Adds a new font to the system.

C#

```
public AddFont(Collection data);
```

Visual Basic

```
Public Sub AddFont(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Font data as a value key pair

Font.DeleteFont Method

Deletes a font from the system.

C#

```
public DeleteFont(Collection data);
```

Visual Basic

```
Public Sub DeleteFont(ByVal data As Collection)
```

Parameters

Parameters	Description
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.

C#

```
public FontData GetAllFonts();
```

Visual Basic

```
Public Function GetAllFonts() As FontData
```

Returns

Ektron.Cms.FontData()

Remarks

User login is required to perform this operation.

Font.GetFont Method

Loads the required font using the font's ID.

C#

```
public FontData GetFont(int Id);
```

Visual Basic

```
Public Function GetFont(ByVal Id As Integer) As FontData
```

Parameters

Parameters	Description
Id	The font's ID

Returns

Integer

Font.GetFontConfigList Method

Returns all of the font details in a drop down control.

C#

```
public String GetFontConfigList();
```

Visual Basic

```
Public Function GetFontConfigList() As String
```

Returns

String

Font.UpdateFont Method

Updates the font.

C#

```
public UpdateFont(Collection data);
```

Visual Basic

```
Public Sub UpdateFont(ByVal data As Collection)
```

Parameters

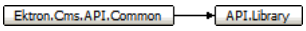
Parameters	Description
data	The font's data

Remarks

A user with administrator privileges must be logged in to perform this operation.

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 API.Library.

Members

Library Methods

	Name	Description
	AddLibraryItem (see page 477)	Adds a library item.
	AddLoadBalanceItem (see page 478)	Adds a load balance path for files and images.
	DeleteLibraryItem (see page 478)	Deletes the library item by extension.
	DeleteLibraryItemById (see page 478)	Deletes the library items from the system.
	DeleteLoadBalanceItem (see page 479)	Deletes the load balance item from the system.

	GetAllChildLibItems (see page 479)	Loads all of the library items by type.
	GetAllLoadBalancePaths (see page 479)	Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.
	GetLibraryItem (see page 480)	Loads the library item's data. This method is overloaded.
	GetLibrarySettings (see page 481)	Loads the library settings.
	GetLibraryTypes (see page 482)	Loads the library types.
	GetLoadBalancePath (see page 482)	Gets the load balance path by ID.
	InternalLibrarySearch (see page 482)	Searches library items.
	UpdateLibrary (see page 483)	Updates a library item.
	UpdateLibrarySettings (see page 483)	Update the library settings.
	UpdateLoadBalanceSettings (see page 483)	Updates the load balance settings.
	UpdateQlinkTemplates (see page 484)	Updates the Quicklink template.

Topics

Name	Description
Library Methods (see page 477)	The methods of the Library class are listed here.

Legend

	Method

Library Methods

	Name	Description
	AddLibraryItem (see page 477)	Adds a library item.
	AddLoadBalanceItem (see page 478)	Adds a load balance path for files and images.
	DeleteLibraryItem (see page 478)	Deletes the library item by extension.
	DeleteLibraryItemById (see page 478)	Deletes the library items from the system.
	DeleteLoadBalanceItem (see page 479)	Deletes the load balance item from the system.
	GetAllChildLibItems (see page 479)	Loads all of the library items by type.
	GetAllLoadBalancePaths (see page 479)	Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.
	GetLibraryItem (see page 480)	Loads the library item's data. This method is overloaded.
	GetLibrarySettings (see page 481)	Loads the library settings.
	GetLibraryTypes (see page 482)	Loads the library types.
	GetLoadBalancePath (see page 482)	Gets the load balance path by ID.
	InternalLibrarySearch (see page 482)	Searches library items.
	UpdateLibrary (see page 483)	Updates a library item.
	UpdateLibrarySettings (see page 483)	Update the library settings.
	UpdateLoadBalanceSettings (see page 483)	Updates the load balance settings.

	UpdateQlinkTemplates (see page 484)	Updates the Quicklink template.

Legend

	Method

Library Methods

The methods of the Library class are listed here.

Public Methods

	Name	Description
	AddLibraryItem (see page 477)	Adds a library (see page 475) item.
	AddLoadBalanceItem (see page 478)	Adds a load balance path for files and images.
	DeleteLibraryItem (see page 478)	Deletes the library (see page 475) item by extension.
	DeleteLibraryItemById (see page 478)	Deletes the library (see page 475) items from the system.
	DeleteLoadBalanceItem (see page 479)	Deletes the load balance item from the system.
	GetAllChildLibItems (see page 479)	Loads all of the library (see page 475) items by type.
	GetAllLoadBalancePaths (see page 479)	Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.
	GetLibraryItem (see page 480)	Loads the library item's data. This method is overloaded.
	GetLibrarySettings (see page 481)	Loads the library (see page 475) settings.
	GetLibraryTypes (see page 482)	Loads the library (see page 475) types.
	GetLoadBalancePath (see page 482)	Gets the load balance path by ID.
	InternalLibrarySearch (see page 482)	Searches library (see page 475) items.
	UpdateLibrary (see page 483)	Updates a library (see page 475) item.
	UpdateLibrarySettings (see page 483)	Update the library (see page 475) settings.
	UpdateLoadBalanceSettings (see page 483)	Updates the load balance settings.
	UpdateQlinkTemplates (see page 484)	Updates the Quicklink template.

Legend

	Method

Library.AddLibraryItem Method

Adds a library (see page 475) item.

C#

```
public int AddLibraryItem(LibraryData libraryData);
```

Visual Basic

```
Public Function AddLibraryItem(ByRef libraryData As LibraryData) As Integer
```

Parameters

Parameters	Description
libraryData	Details of the new libray item

Remarks

A user with administrator privileges must be logged in to perform this operation.

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
Type	The path type, either images or files
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 (see page 475) item by extension.

C#

```
public DeleteLibraryItem(Collection pagedata);
```

Visual Basic

```
Public Sub DeleteLibraryItem(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
pagedata	Library (see page 475) item data

Remarks

A user with administrator privileges must be logged in to perform this operation.

Library.DeleteLibraryItemById Method

Deletes the library (see page 475) items from the system.

C#

```
public DeleteLibraryItemById(Collection pagedata);
```

Visual Basic

```
Public Sub DeleteLibraryItemById(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
pagedata	Library (see page 475) item data

Remarks

User login with administrator privileges are required to perform this operation.

Library.DeleteLoadBalanceItem Method

Deletes the load balance item from the system.

C#

```
public DeleteLoadBalanceItem(int Id);
```

Visual Basic

```
Public Sub DeleteLoadBalanceItem(ByVal Id As Integer)
```

Parameters

Parameters	Description
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 (see page 475) items by type.

C#

```
public LibraryData GetAllChildLibItems(String Type, int ParentId, String OrderBy, int
currentPageNum, int pageSize, int totalPages);
```

Visual Basic

```
Public Function GetAllChildLibItems(ByVal Type As String, ByVal ParentId As Integer, ByVal
OrderBy As String, ByVal currentPageNum As Integer, ByVal pageSize As Integer, ByRef
totalPages As Integer) As LibraryData
```

Parameters

Parameters	Description
Type	The type (Required)
ParentId	The folder's ID
OrderBy	The order to load the library (see page 475) items. You can pass librarytitle, libraryid, libraryfilename, date as string.
currentPageNum	The current page number
pageSize	The page size
totalPages	The total pages

Returns

Ektron.Cms.LibraryData()

Remarks

User login required to perform this operation.

GetAllLoadBalancePaths Method

Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.

Overload List

	Name	Description
	Library.GetAllLoadBalancePaths (Integer, String) (see page 480)	Returns an array of LoadBalanceData.

	Library.GetAllLoadBalancePaths (String) (see page 480)	Returns all the load balance paths as a array of LoadBalanceData.

Library.GetAllLoadBalancePaths Method (Integer, String)

Returns an array of LoadBalanceData.

C#

```
public LoadBalanceData GetAllLoadBalancePaths(int Id, String Type);
```

Visual Basic

```
Public Function GetAllLoadBalancePaths(ByVal Id As Integer, ByVal Type As String) As
LoadBalanceData
```

Parameters

Parameters	Description
Id	Load balance ID
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.

C#

```
public LoadBalanceData GetAllLoadBalancePaths(String Type);
```

Visual Basic

```
Public Function GetAllLoadBalancePaths(ByVal Type As String) As LoadBalanceData
```

Parameters

Parameters	Description
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

Loads the library item's data. This method is overloaded.

Overload List

	Name	Description
	Library.GetLibraryItem (Integer) (see page 481)	Loads the library (see page 475) item data.

	Library.GetLibraryItem (Integer, Integer) (see page 481)	Loads the library (see page 475) item's data.

Library.GetLibraryItem Method (Integer)

Loads the library (see page 475) item data.

C#

```
public LibraryData GetLibraryItem(int LibID);
```

Visual Basic

```
Public Function GetLibraryItem(ByVal LibID As Integer) As LibraryData
```

Parameters

Parameters	Description
LibID	The library (see page 475) ID

Returns

Ektron.Cms.LibraryData

Library.GetLibraryItem Method (Integer, Integer)

Loads the library (see page 475) item's data.

C#

```
public LibraryData GetLibraryItem(int Id, int ParentId);
```

Visual Basic

```
Public Function GetLibraryItem(ByVal Id As Integer, ByVal ParentId As Integer) As LibraryData
```

Parameters

Parameters	Description
Id	The library (see page 475)'s ID
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 (see page 475) settings.

C#

```
public LibraryConfigData GetLibrarySettings(int Id);
```

Visual Basic

```
Public Function GetLibrarySettings(Optional ByVal Id As Integer = -1) As LibraryConfigData
```

Parameters

Parameters	Description
Id	The folder's ID

Returns

Ektron.Cms.LibraryConfigData

Remarks

Login required to perform this operation.

Library.GetLibraryTypes Method

Loads the library (see page 475) types.

C#

```
public LibraryTypeData GetLibraryTypes(int Id, String OrderBy);
```

Visual Basic

```
Public Function GetLibraryTypes(ByVal Id As Integer, ByVal OrderBy As String) As
LibraryTypeData
```

Parameters

Parameters	Description
Id	The folder's ID
OrderBy	Order by clause

Returns

Object

Remarks

Library (see page 475) read permission is required to perform this operation.

Library.GetLoadBalancePath Method

Gets the load balance path by ID.

C#

```
public LoadBalanceData GetLoadBalancePath(int Id);
```

Visual Basic

```
Public Function GetLoadBalancePath(ByVal Id As Integer) As LoadBalanceData
```

Parameters

Parameters	Description
Id	the load balance path's ID

Returns

Ektron.Cms.LoadBalanceData

Library.InternalLibrarySearch Method

Searches library (see page 475) items.

C#

```
public SearchResultData InternalLibrarySearch(Collection pagedata,
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
pagedata	Search criteria
sar	Search asset criteria

Returns

Ektron.Cms.SearchResultData()

Remarks

User login is required to perform this operation.

Library.UpdateLibrary Method

Updates a library (see page 475) item.

C#

```
public UpdateLibrary(Collection pagedata);
```

Visual Basic

```
Public Sub UpdateLibrary(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
pagedata	Library (see page 475) item data

Library.UpdateLibrarySettings Method

Update the library (see page 475) settings.

C#

```
public UpdateLibrarySettings(Collection pagedata);
```

Visual Basic

```
Public Sub UpdateLibrarySettings(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
pagedata	Library (see page 475) settings data

Remarks

A user with administrator privileges must be logged in to perform this operation.

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
Id	The load balance data
pagedata	The load balance items

Remarks

A user with administrator privileges must be logged in to perform this operation.

Library.UpdateQlinkTemplates Method

Updates the Quicklink template.

C#

```
public UpdateQlinkTemplates(Collection pagedata);
```

Visual Basic

```
Public Sub UpdateQlinkTemplates(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
pagedata	Quicklink template data

Remarks

A user with administrator privileges must be logged in to perform this operation.

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 API.Localization.

Members

Localization Methods

	Name	Description
	GetAddViewLanguage (see page 487)	Returns an array of all the active content (see page 337) languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") for the specified content (see page 337) ID or whether it does not exist in that language (Type="ADD").
	GetAddViewLanguageForAllContent (see page 487)	Returns an array of all the active content (see page 337) languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one content (see page 337) block or whether no content (see page 337) exists in that language (Type="ADD").
	GetAddViewLanguageForMenus (see page 488)	Returns an array of all the active content (see page 337) 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 488)	Returns a list of content (see page 337) items (i.e., XLIFF skeletons) associated with a given job.
	GetFile (see page 488)	Returns the specified file record.
	GetFiles (see page 489)	Returns a list of files associated with a given job.
	GetFilesInZipFile (see page 489)	Returns a list of files in a given ZIP file.

	GetFlagUrl (see page 489)	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.
	GetInterchangeFile (see page 491)	Returns the interchange (i.e., XLIFF) file record specified by IDID.
	GetInterchangeFiles (see page 491)	Returns a list of interchange files (i.e., XLIFF files) associated with a given content (see page 337) item (i.e., XLIFF Skeleton).
	GetInterchangeFilesInZipFile (see page 492)	Returns a list of interchange files (i.e., XLIFF files) contained within a given ZIP file.
	GetJobs (see page 492)	Gets a list of top level jobs or a list of the sub jobs of the given job. This method is overloaded.
	GetLocalizationUrl (see page 493)	Returns a URL to the base directory where localization files are stored. `For example, /CMS400Demo/uploadedfiles/localization.`
	GetTranslationUploadDirectory (see page 493)	Returns the server directory path to upload XLIFF and ZIP files for importing. For example, `C:\inetpub\wwwroot\CMS400Demo\workarea\uploadedfiles\localization`. See Also: StartImportTranslation.
	StartExportContentForTranslation (see page 493)	Starts the process of exporting the content (see page 337) 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 494)	Starts the process of exporting all the content (see page 337) specified by `ContentType` in the specified folder. Optionally, all the content (see page 337) 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 494)	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 495)	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.

Topics

Name	Description
Localization Methods (see page 486)	The methods of the Localization class are listed here.

Legend

	Method

Localization Methods

	Name	Description
	GetAddViewLanguage (see page 487)	Returns an array of all the active content (see page 337) languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") for the specified content (see page 337) ID or whether it does not exist in that language (Type="ADD").
	GetAddViewLanguageForAllContent (see page 487)	Returns an array of all the active content (see page 337) languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one content (see page 337) block or whether no content (see page 337) exists in that language (Type="ADD").
	GetAddViewLanguageForMenus (see page 488)	Returns an array of all the active content (see page 337) 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 488)	Returns a list of content (see page 337) items (i.e., XLIFF skeletons) associated with a given job.
	GetFile (see page 488)	Returns the specified file record.
	GetFiles (see page 489)	Returns a list of files associated with a given job.
	GetFilesInZipFile (see page 489)	Returns a list of files in a given ZIP file.
	GetFlagUrl (see page 489)	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.
	GetInterchangeFile (see page 491)	Returns the interchange (i.e., XLIFF) file record specified by IDID.
	GetInterchangeFiles (see page 491)	Returns a list of interchange files (i.e., XLIFF files) associated with a given content (see page 337) item (i.e., XLIFF Skeleton).
	GetInterchangeFilesInZipFile (see page 492)	Returns a list of interchange files (i.e., XLIFF files) contained within a given ZIP file.
	GetJobs (see page 492)	Gets a list of top level jobs or a list of the sub jobs of the given job. This method is overloaded.

	GetLocalizationUrl (see page 493)	Returns a URL to the base directory where localization files are stored. `For example, /CMS400Demo/uploadedfiles/localization.`
	GetTranslationUploadDirectory (see page 493)	Returns the server directory path to upload XLIFF and ZIP files for importing. For example, `C:\inetpub\wwwroot\CMS400Demo\workarea\uploadedfiles\localization`. See Also: StartImportTranslation.
	StartExportContentForTranslation (see page 493)	Starts the process of exporting the content (see page 337) 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 494)	Starts the process of exporting all the content (see page 337) specified by `ContentType` in the specified folder. Optionally, all the content (see page 337) 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 494)	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 495)	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.

Legend

	Method

Localization Methods

The methods of the Localization class are listed here.

Public Methods

	Name	Description
	GetAddViewLanguage (see page 487)	Returns an array of all the active content (see page 337) languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") for the specified content (see page 337) ID or whether it does not exist in that language (Type="ADD").
	GetAddViewLanguageForAllContent (see page 487)	Returns an array of all the active content (see page 337) languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one content (see page 337) block or whether no content (see page 337) exists in that language (Type="ADD").
	GetAddViewLanguageForMenus (see page 488)	Returns an array of all the active content (see page 337) 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 488)	Returns a list of content (see page 337) items (i.e., XLIFF skeletons) associated with a given job.
	GetFile (see page 488)	Returns the specified file record.
	GetFiles (see page 489)	Returns a list of files associated with a given job.
	GetFilesInZipFile (see page 489)	Returns a list of files in a given ZIP file.
	GetFlagUrl (see page 489)	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.
	GetInterchangeFile (see page 491)	Returns the interchange (i.e., XLIFF) file record specified by IDID.
	GetInterchangeFiles (see page 491)	Returns a list of interchange files (i.e., XLIFF files) associated with a given content (see page 337) item (i.e., XLIFF Skeleton).
	GetInterchangeFilesInZipFile (see page 492)	Returns a list of interchange files (i.e., XLIFF files) contained within a given ZIP file.
	GetJobs (see page 492)	Gets a list of top level jobs or a list of the sub jobs of the given job. This method is overloaded.
	GetLocalizationUrl (see page 493)	Returns a URL to the base directory where localization (see page 484) files are stored. `For example, /CMS400Demo/uploadedfiles/localization (see page 484).`
	GetTranslationUploadDirectory (see page 493)	Returns the server directory path to upload XLIFF and ZIP files for importing. For example, `C:\inetpub\wwwroot\CMS400Demo\workarea\uploadedfiles\localization (see page 484).` See Also: StartImportTranslation.

	StartExportContentForTranslation (see page 493)	Starts the process of exporting the content (see page 337) 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 494)	Starts the process of exporting all the content (see page 337) specified by `ContentType` in the specified folder. Optionally, all the content (see page 337) 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 494)	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 495)	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.

Legend

	Method

Localization.GetAddViewLanguage Method

Returns an array of all the active content (see page 337) languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") for the specified content (see page 337) ID or whether it does not exist in that language (Type="ADD").

C#

```
public LanguageData GetAddViewLanguage(int Id);
```

Visual Basic

```
Public Function GetAddViewLanguage(ByVal Id As Integer) As LanguageData
```

Parameters

Parameters	Description
`Id`	Content (see page 337) ID

Returns

Ektron.Cms.LanguageData()

Localization.GetAddViewLanguageForAllContent Method

Returns an array of all the active content (see page 337) languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one content (see page 337) block or whether no content (see page 337) 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.

See Also

Ektron.Cms.LanguageData

Localization.GetAddViewLanguageForMenus Method

Returns an array of all the active content (see page 337) 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.

See Also

Ektron.Cms.LanguageData

Localization.GetContentItems Method

Returns a list of content (see page 337) items (i.e., XLIFF skeletons) associated with a given job.

C#

```
public DS.LocalizationSkeletonDataTable GetContentItems(int JobID);
```

Visual Basic

```
Public Function GetContentItems(ByVal JobID As Integer) As DS.LocalizationSkeletonDataTable
```

Parameters

Parameters	Description
JobID	Specifies the job.

Returns

A LocalizationJobDataSet.LocalizationSkeletonDataTable data table of content (see page 337) (i.e., skeleton) items.

Localization.GetFile Method

Returns the specified file record.

C#

```
public DS.LocalizationJobFileRow GetFile(int FileID);
```

Visual Basic

```
Public Function GetFile(ByVal FileID As Integer) As DS.LocalizationJobFileRow
```

Parameters

Parameters	Description
FileID	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(int JobID);
```

Visual Basic

```
Public Function GetFiles(ByVal JobID As Integer) As DS.LocalizationJobFileDataTable
```

Parameters

Parameters	Description
JobID	Specifies the job.

Returns

A LocalizationJobDataSet.LocalizationJobFileDataTable data table 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(int FileID);
```

Visual Basic

```
Public Function GetFilesInZipFile(ByVal FileID As Integer) As DS.LocalizationJobFileDataTable
```

Parameters

Parameters	Description
FileID	Specifies a file (e.g., ZIP file)

Returns

A LocalizationJobDataSet.LocalizationJobFileDataTable data table 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

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 490)	Returns a URL to a small image of a national flag for the country and language specified by LanguageID.

	Localization.GetFlagUrl (LanguageData) (see page 490)	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 490)	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
LanguageID	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
Data	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
Data	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 IDID.

C#

```
public DS.LocalizationTargetRow GetInterchangeFile(int ID);
```

Visual Basic

```
Public Function GetInterchangeFile(ByVal ID As Integer) As DS.LocalizationTargetRow
```

Parameters

Parameters	Description
ID	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 (see page 337) item (i.e., XLIFF Skeleton).

C#

```
public DS.LocalizationTargetDataTable GetInterchangeFiles(int ItemID);
```

Visual Basic

```
Public Function GetInterchangeFiles(ByVal ItemID As Integer) As DS.LocalizationTargetDataTable
```

Parameters

Parameters	Description
ItemID	Specifies the content (see page 337) item (i.e., XLIFF skeleton).

Returns

A LocalizationJobDataSet.LocalizationTargetDataTable data table 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 (see page 337) item it was generated from (the original language) and a new skeleton content (see page 337) item generated by merging the XLIFF with the original skeleton to produce a skeleton in the target language. A skeleton consists of the content (see page 337), typically the main HTML content (see page 337) 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(int FileID);
```

Visual Basic

```
Public Function GetInterchangeFilesInZipFile(ByVal FileID As Integer) As
DS.LocalizationTargetDataTable
```

Parameters

Parameters	Description
FileID	Specifies the ZIP file.

Returns

A LocalizationJobDataSet.LocalizationTargetDataTable data table of XLIFF files.

Remarks

XLIFF is a standard to interchange localization (see page 484) information for translation of text. The XLIFF files are compressed together into one or more ZIP files for convenience.

GetJobs Method

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 492)	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 (Integer)
	Localization.GetJobs (Integer) (see page 493)	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 (Integer)

C#

```
public DS.LocalizationJobDataTable GetJobs();
```

Visual Basic

```
Public Function GetJobs() As DS.LocalizationJobDataTable
```

Returns

A LocalizationJobDataSet.LocalizationJobDataTable data table of top-level jobs.

Localization.GetJobs Method (Integer)

Returns a list of the sub jobs of the given job.

This function is overloaded. See Also: GetJobs

C#

```
public DS.LocalizationJobDataTable GetJobs(int ParentJobID);
```

Visual Basic

```
Public Function GetJobs(ByVal ParentJobID As Integer) As DS.LocalizationJobDataTable
```

Parameters

Parameters	Description
ParentJobID	Specifies the job.

Returns

A LocalizationJobDataSet.LocalizationJobDataTable data table of jobs.

Localization.GetLocalizationUrl Method

Returns a URL to the base directory where localization (see page 484) files are stored. For example, /CMS400Demo/uploadedfiles/localization (see page 484).

C#

```
public String GetLocalizationUrl();
```

Visual Basic

```
Public Function GetLocalizationUrl() As String
```

Returns

A URL to the base localization (see page 484) directory.

Localization.GetTranslationUploadDirectory Method

Returns the server directory path to upload XLIFF and ZIP files for importing. For example, C:\inetpub\wwwroot\CMS400Demo\workarea\uploadedfiles\localization (see page 484).

See Also: StartImportTranslation.

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 (see page 337) 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.

C#

```
public StartExportContentForTranslation(String ListOfContentIDs, String ListOfLanguageIDs);
```

Visual Basic

```
Public Sub StartExportContentForTranslation(ByVal ListOfContentIDs As String, Optional ByVal
ListOfLanguageIDs As String = "")
```

Parameters

Parameters	Description
ListOfContentIDs	Comma-delimited list of content (see page 337) IDs. e.g., "11,12"
ListOfLanguageIDs	Comma-delimited list of language IDs. e.g., "1031,1036"

Remarks

To export all content (see page 337) in a folder, use StartExportFolderForTranslation. See http://www.xliff.org for more information about XLIFF.

Localization.StartExportFolderForTranslation Method

Starts the process of exporting all the content (see page 337) specified by ContentType in the specified folder. Optionally, all the content (see page 337) 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.

C#

```
public StartExportFolderForTranslation(int FolderID, int ContentType, Boolean
IncludeSubfolders, String ListOfLanguageIDs);
```

Visual Basic

```
Public Sub StartExportFolderForTranslation(ByVal FolderID As Integer, ByVal ContentType As
Integer, ByVal IncludeSubfolders As Boolean, Optional ByVal ListOfLanguageIDs As String = "")
```

Parameters

Parameters	Description
FolderID	Specifies the folder containing the content (see page 337) to export.
ContentType	Specifies which type or types of content (see page 337) to export.
IncludeSubfolders	Specifies whether to export just the specified folder or all subfolders as well.
ListOfLanguageIDs	Comma-delimited list of language IDs. e.g., "1031,1036"

Remarks

To export just selected content (see page 337), use StartExportFolderForTranslation.

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 method.

C#

```
public StartExportMenusForTranslation(String ListOfLanguageIDs);
```

Visual Basic

```
Public Sub StartExportMenusForTranslation(Optional ByVal ListOfLanguageIDs As String = "")
```

Parameters

Parameters	Description
ListOfLanguageIDs	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. 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.

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 API.Metadata.

Members

Metadata Methods

	Name	Description
	AddMetaDataType (see page 496)	
	DeleteMetadataType (see page 497)	Deletes a metadata type from the system.
	GetContentMetadataList (see page 498)	
	GetMetadataType (see page 499)	Loads all content (see page 337) metadata.
	GetMetaDataTypes (see page 499)	Loads all the metadata types.
	UpdateMetadataType (see page 499)	Updates the metadata type.

Topics

Name	Description
Metadata Methods (see page 496)	The methods of the Metadata class are listed here.

Legend

	Method

Metadata Methods

	Name	Description
	AddMetaDataType (see page 496)	
	DeleteMetadataType (see page 497)	Deletes a metadata type from the system.
	GetContentMetadataList (see page 498)	
	GetMetadataType (see page 499)	Loads all content (see page 337) metadata.
	GetMetaDataTypes (see page 499)	Loads all the metadata types.
	UpdateMetadataType (see page 499)	Updates the metadata type.

Legend

	Method

Metadata Methods

The methods of the Metadata class are listed here.

Public Methods

	Name	Description
	AddMetaDataType (see page 496)	
	DeleteMetadataType (see page 497)	Deletes a metadata (see page 495) type from the system.
	GetContentMetadataList (see page 498)	
	GetMetadataType (see page 499)	Loads all content (see page 337) metadata (see page 495).
	GetMetaDataTypes (see page 499)	Loads all the metadata (see page 495) types.
	UpdateMetadataType (see page 499)	Updates the metadata (see page 495) type.

Legend

	Method

AddMetaDataType Method

Overload List

	Name	Description
	Metadata.AddMetaDataType (Collection) (see page 497)	Adds a new metadata (see page 495) type to the system
	Metadata.AddMetaDataType (ContentMetaData) (see page 497)	Adds a new metadata (see page 495) type to the system

Metadata.AddMetaDataType Method (Collection)

Adds a new metadata (see page 495) type to the system

C#

```
public AddMetaDataType(Collection pagedata);
```

Visual Basic

```
Public Sub AddMetaDataType(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
pagedata	Metadata (see page 495) items as value key pair

Remarks

A user with administrator privileges must be logged in to perform this operation.

Metadata.AddMetaDataType Method (ContentMetaData)

Adds a new metadata (see page 495) type to the system

C#

```
public AddMetaDataType(ContentMetaData metaDataItem);
```

Visual Basic

```
Public Sub AddMetaDataType(ByRef metaDataItem As ContentMetaData)
```

Parameters

Parameters	Description
metaDataItem	ContentMetadata object containing metadata (see page 495) definition details

Remarks

A user with administrator privileges must be logged in to perform this operation.

Metadata.DeleteMetadataType Method

Deletes a metadata (see page 495) type from the system.

C#

```
public DeleteMetadataType(int Id);
```

Visual Basic

```
Public Sub DeleteMetadataType(ByVal Id As Integer)
```

Parameters

Parameters	Description
Id	Metatype ID

Remarks

A user with administrator privileges must be logged in to perform this operation.

GetContentMetadataList Method

Overload List

	Name	Description
	Metadata.GetContentMetadataList (Integer) (see page 498)	Loads all of the metadata (see page 495) for the given content (see page 337).
	Metadata.GetContentMetadataList (Integer, Boolean) (see page 498)	Loads all of the metadata (see page 495) for the given content (see page 337).

Metadata.GetContentMetadataList Method (Integer)

Loads all of the metadata (see page 495) for the given content (see page 337).

C#

```
public CustomAttributeList GetContentMetadataList(int ContentId);
```

Visual Basic

```
Public Function GetContentMetadataList(ByVal ContentId As Integer) As CustomAttributeList
```

Parameters

Parameters	Description
ContentId	The contents ID.

Returns

Ektron.Cms.CustomAttributeList

Remarks

Read permissions required.

Metadata.GetContentMetadataList Method (Integer, Boolean)

Loads all of the metadata (see page 495) for the given content (see page 337).

C#

```
public CustomAttributeList GetContentMetadataList(int ContentId, Boolean preview);
```

Visual Basic

```
Public Function GetContentMetadataList(ByVal ContentId As Integer, ByVal preview As Boolean)
As CustomAttributeList
```

Parameters

Parameters	Description
ContentId	The contents ID.
preview	True implies get metadata (see page 495) from staging content (see page 337), else use published content (see page 337)

Returns

Ektron.Cms.CustomAttributeList

Remarks

Read permissions required.

Metadata.GetMetadataType Method

Loads all content (see page 337) metadata (see page 495).

C#

```
public ContentMetaData GetMetadataType(int Id);
```

Visual Basic

```
Public Function GetMetadataType(ByVal Id As Integer) As ContentMetaData
```

Parameters

Parameters	Description
Id	The metadata (see page 495) 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 (see page 495) types.

C#

```
public ContentMetaData GetMetaDataTypes(String OrderBy);
```

Visual Basic

```
Public Function GetMetaDataTypes(ByVal OrderBy As String) As ContentMetaData
```

Parameters

Parameters	Description
OrderBy	Order by clause

Returns

Ektron.Cms.ContentMetaData()

Remarks

User login is required to perform this operation.

Metadata.UpdateMetadataType Method

Updates the metadata (see page 495) type.

C#

```
public UpdateMetadataType(Collection pagedata);
```

Visual Basic

```
Public Sub UpdateMetadataType(ByVal pagedata As Collection)
```

Parameters

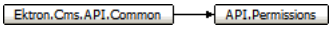
Parameters	Description
pagedata	Metadata (see page 495) types

Remarks

A user with administrator privileges must be logged in to perform this operation.

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 API.Permissions.

Members

Permissions Methods

	Name	Description
	AddItemPermission (see page 501)	Adds the item permissions for a user or a usergroup.
	DeleteItemPermission (see page 502)	Deletes a user or a group from the item's permissions
	DisableContentInheritance (see page 502)	Disable permissions inheritance on the content (see page 337) and copies current permissions to the content (see page 337).
	DisableFolderInheritance (see page 502)	Disable permissions inheritance on the folder and copies current permissions to the folder.
	EnableContentInheritance (see page 503)	Deletes current permissions from the content (see page 337) and enable permissions inheritance on the content (see page 337).
	EnableFolderInheritance (see page 503)	Deletes current permissions from the folder and enable permissions inheritance on the folder.
	GetUserPermissions (see page 503)	
	LoadPermissions (see page 504)	Loads the permission for given object, such as content (see page 337), folder, task, etc.
	UpdateItemPermission (see page 505)	Updates the item permissions for a user or a usergroup.

Topics

Name	Description
Permissions Methods (see page 501)	The methods of the Permissions class are listed here.

Legend

	Method

Permissions Methods

	Name	Description
	AddItemPermission (see page 501)	Adds the item permissions for a user or a usergroup.
	DeleteItemPermission (see page 502)	Deletes a user or a group from the item's permissions
	DisableContentInheritance (see page 502)	Disable permissions inheritance on the content (see page 337) and copies current permissions to the content (see page 337).
	DisableFolderInheritance (see page 502)	Disable permissions inheritance on the folder and copies current permissions to the folder.

	EnableContentInheritance (see page 503)	Deletes current permissions from the content (see page 337) and enable permissions inheritance on the content (see page 337).
	EnableFolderInheritance (see page 503)	Deletes current permissions from the folder and enable permissions inheritance on the folder.
	GetUserPermissions (see page 503)	
	LoadPermissions (see page 504)	Loads the permission for given object, such as content (see page 337), folder, task, etc.
	UpdateItemPermission (see page 505)	Updates the item permissions for a user or a usergroup.

Legend

	Method

Permissions Methods

The methods of the Permissions class are listed here.

Public Methods

	Name	Description
	AddItemPermission (see page 501)	Adds the item permissions (see page 500) for a user or a usergroup.
	DeleteItemPermission (see page 502)	Deletes a user or a group from the item's permissions (see page 500)
	DisableContentInheritance (see page 502)	Disable permissions (see page 500) inheritance on the content (see page 337) and copies current permissions (see page 500) to the content (see page 337).
	DisableFolderInheritance (see page 502)	Disable permissions (see page 500) inheritance on the folder and copies current permissions (see page 500) to the folder.
	EnableContentInheritance (see page 503)	Deletes current permissions (see page 500) from the content (see page 337) and enable permissions (see page 500) inheritance on the content (see page 337).
	EnableFolderInheritance (see page 503)	Deletes current permissions (see page 500) from the folder and enable permissions (see page 500) inheritance on the folder.
	GetUserPermissions (see page 503)	
	LoadPermissions (see page 504)	Loads the permission for given object, such as content (see page 337), folder, task, etc.
	UpdateItemPermission (see page 505)	Updates the item permissions (see page 500) for a user or a usergroup.

Legend

	Method

Permissions.AddItemPermission Method

Adds the item permissions (see page 500) 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
userPermData	User's Permissions (see page 500) Data

Returns

False if it's successful

Permissions.DeleteItemPermission Method

Deletes a user or a group from the item's permissions (see page 500)

C#

```
public Boolean DeleteItemPermission(int id, Boolean IsGroupId, int itemId,
Ektron.Cms.Common.EkEnumeration.CMSObjectTypes itemType);
```

Visual Basic

```
Public Function DeleteItemPermission(ByVal id As Integer, ByVal IsGroupId As Boolean, ByVal
itemId As Integer, ByVal itemType As Ektron.Cms.Common.EkEnumeration.CMSObjectTypes) As Boolean
```

Parameters

Parameters	Description
id	User or Group Id
IsGroupId	Given ID is a user id or a group id?
itemId	ContentId or FolderId
itemType	Given item id is EkEnumeration.CMSObjectTypes.Content (see page 337) or EkEnumeration.CMSObjectTypes.Folder?

Returns

False if it's successful.

Permissions.DisableContentInheritance Method

Disable permissions (see page 500) inheritance on the content (see page 337) and copies current permissions (see page 500) to the content (see page 337).

C#

```
public Boolean DisableContentInheritance(int id);
```

Visual Basic

```
Public Function DisableContentInheritance(ByVal id As Integer) As Boolean
```

Parameters

Parameters	Description
id	Content (see page 337) Id

Returns

False if it's successful.

Permissions.DisableFolderInheritance Method

Disable permissions (see page 500) inheritance on the folder and copies current permissions (see page 500) to the folder.

C#

```
public Boolean DisableFolderInheritance(int id);
```

Visual Basic

```
Public Function DisableFolderInheritance(ByVal id As Integer) As Boolean
```

Parameters

Parameters	Description
id	folder Id

Returns

False if it's successful.

Permissions.EnableContentInheritance Method

Deletes current permissions (see page 500) from the content (see page 337) and enable permissions (see page 500) inheritance on the content (see page 337).

C#

```
public Boolean EnableContentInheritance(int id);
```

Visual Basic

```
Public Function EnableContentInheritance(ByVal id As Integer) As Boolean
```

Parameters

Parameters	Description
id	Content (see page 337) Id

Returns

False if it's successful.

Permissions.EnableFolderInheritance Method

Deletes current permissions (see page 500) from the folder and enable permissions (see page 500) inheritance on the folder.

C#

```
public Boolean EnableFolderInheritance(int id);
```

Visual Basic

```
Public Function EnableFolderInheritance(ByVal id As Integer) As Boolean
```

Parameters

Parameters	Description
id	folder Id

Returns

False if it's successful.

GetUserPermissions Method

Overload List

	Name	Description
	Permissions.GetUserPermissions (Integer, String, Integer, String) (see page 503)	Loads the user permissions (see page 500) as a array of UserPermissionData.
	Permissions.GetUserPermissions (Integer, String, Integer, String, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) (see page 504)	Loads the user permissions (see page 500) as a array of UserPermissionData.

Permissions.GetUserPermissions Method (Integer, String, Integer, String)

Loads the user permissions (see page 500) as a array of UserPermissionData.

C#

```
public UserPermissionData GetUserPermissions(int Id, String ItemType, int UserId, String
UserGroupList);
```

Visual Basic

```
Public Function GetUserPermissions(ByVal Id As Integer, ByVal ItemType As String, ByVal UserId
As Integer, ByVal UserGroupList As String) As UserPermissionData
```

Parameters

Parameters	Description
Id	the item's ID
ItemType	The item's type
UserId	The user's ID
UserGroupList	the group list

Returns

Ektron.Cms.UserPermissionData()

Permissions.GetUserPermissions Method (Integer, String, Integer, String, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType)

Loads the user permissions (see page 500) as a array of UserPermissionData.

C#

```
public UserPermissionData GetUserPermissions(int Id, String ItemType, int UserId, String
UserGroupList, ContentAPI.PermissionUserType PermissionType, ContentAPI.PermissionRequestType
PermissionRequest);
```

Visual Basic

```
Public Function GetUserPermissions(ByVal Id As Integer, ByVal ItemType As String, ByVal UserId
As Integer, ByVal UserGroupList As String, ByVal PermissionType As
ContentAPI.PermissionUserType, ByVal PermissionRequest As ContentAPI.PermissionRequestType) As
UserPermissionData
```

Parameters

Parameters	Description
Id	the item's ID
ItemType	The item's type
UserId	The user's ID
UserGroupList	the group list
PermissionType	PermissionUserType
PermissionRequest	PermissionRequestType

Returns

Ektron.Cms.UserPermissionData()

Permissions.LoadPermissions Method

Loads the permission for given object, such as content (see page 337), folder, task, etc.

C#

```
public PermissionData LoadPermissions(int Id, String RequestType,
ContentAPI.PermissionResultType Type);
```

Visual Basic

```
Public Function LoadPermissions(ByVal Id As Integer, ByVal RequestType As String, Optional
ByVal Type As ContentAPI.PermissionResultType = ContentAPI.PermissionResultType.Common) As
```

```
PermissionData
```

Parameters

Parameters	Description
Id	The object ID
RequestType	The object type := content (see page 337) | folder | collections | tasks
Type	The permission result type

Returns

Ektron.Cms.PermissionData

Remarks

User login with administrator privileges are required to perform this operation.

Permissions.UpdateItemPermission Method

Updates the item permissions (see page 500) for a user or a usergroup.

C#

```
public Boolean UpdateItemPermission(UserPermissionData userPermData);
```

Visual Basic

```
Public Function UpdateItemPermission(ByVal userPermData As UserPermissionData) As Boolean
```

Parameters

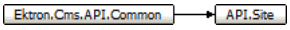
Parameters	Description
userPermData	User's Permissions (see page 500) Data

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 API.Site.

Members

Site Methods

	Name	Description
	BreakTaskPermissionInherit (see page 507)	Removes the task permissions for a given user.

	DeleteCookie (see page 508)	Removes the cookie from the site.
	GetAllActiveLanguages (see page 508)	Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language or Settings area.
	GetAllLanguages (see page 508)	Gets a list of all the available languages.
	GetClientDayNames (see page 508)	Returns the day name of a given content (see page 337) language (or site language).
	GetClientMonthNames (see page 509)	Returns the month name of a given content (see page 337) language (or site language)
	GetCookieValue (see page 509)	Reads the cookie value.
	GetLanguage (see page 509)	Returns the details about the given language ID.
	GetLanguages (see page 510)	Gets a list of all languages that meet a given state, for example, active.
	GetSiteVariables (see page 510)	Returns all of the settings data.
	GetTaskPermission (see page 510)	Returns 1 if the user has one of the task permissions, otherwise 0.
	IsTaskPermissionInherited (see page 511)	Returns true if a user has already granted one of the task permissions.
	SetCookieValue (see page 511)	Resets the cookie value.
	SetTaskPermission (see page 511)	Updates the task permission(s) for the given user or group.
	UpdateSiteEnabledLanguages (see page 512)	Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled.
	UpdateSiteVariables (see page 512)	Update the site settings

Topics

Name	Description
Site Methods (see page 507)	The methods of the Site class are listed here.

Legend

	Method

Site Methods

	Name	Description
	BreakTaskPermissionInherit (see page 507)	Removes the task permissions for a given user.
	DeleteCookie (see page 508)	Removes the cookie from the site.
	GetAllActiveLanguages (see page 508)	Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language or Settings area.
	GetAllLanguages (see page 508)	Gets a list of all the available languages.
	GetClientDayNames (see page 508)	Returns the day name of a given content (see page 337) language (or site language).
	GetClientMonthNames (see page 509)	Returns the month name of a given content (see page 337) language (or site language)
	GetCookieValue (see page 509)	Reads the cookie value.
	GetLanguage (see page 509)	Returns the details about the given language ID.
	GetLanguages (see page 510)	Gets a list of all languages that meet a given state, for example, active.
	GetSiteVariables (see page 510)	Returns all of the settings data.
	GetTaskPermission (see page 510)	Returns 1 if the user has one of the task permissions, otherwise 0.

	IsTaskPermissionInherited (see page 511)	Returns true if a user has already granted one of the task permissions.
	SetCookieValue (see page 511)	Resets the cookie value.
	SetTaskPermission (see page 511)	Updates the task permission(s) for the given user or group.
	UpdateSiteEnabledLanguages (see page 512)	Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled.
	UpdateSiteVariables (see page 512)	Update the site settings

Legend

	Method

Site Methods

The methods of the Site class are listed here.

Public Methods

	Name	Description
	BreakTaskPermissionInherit (see page 507)	Removes the task permissions for a given user.
	DeleteCookie (see page 508)	Removes the cookie from the site.
	GetAllActiveLanguages (see page 508)	Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language or Settings area.
	GetAllLanguages (see page 508)	Gets a list of all the available languages.
	GetClientDayNames (see page 508)	Returns the day name of a given content (see page 337) language (or site language).
	GetClientMonthNames (see page 509)	Returns the month name of a given content (see page 337) language (or site language)
	GetCookieValue (see page 509)	Reads the cookie value.
	GetLanguage (see page 509)	Returns the details about the given language ID.
	GetLanguages (see page 510)	Gets a list of all languages that meet a given state, for example, active.
	GetSiteVariables (see page 510)	Returns all of the settings data.
	GetTaskPermission (see page 510)	Returns 1 if the user has one of the task permissions, otherwise 0.
	IsTaskPermissionInherited (see page 511)	Returns true if a user has already granted one of the task permissions.
	SetCookieValue (see page 511)	Resets the cookie value.
	SetTaskPermission (see page 511)	Updates the task permission(s) for the given user or group.
	UpdateSiteEnabledLanguages (see page 512)	Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled.
	UpdateSiteVariables (see page 512)	Update the site settings

Legend

	Method

Site.BreakTaskPermissionInherit Method

Removes the task permissions for a given user.

C#

```
public BreakTaskPermissionInherit(int Id);
```

Visual Basic

```
Public Sub BreakTaskPermissionInherit(ByVal Id As Integer)
```

Parameters

Parameters	Description
Id	User ID

Remarks

The user should be logged in to the system. Otherwise, the user will get the permission denied error.

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

Site.GetAllLanguages Method

Gets a list of all the available languages.

C#

```
public LanguageData GetAllLanguages(String OrderBy);
```

Visual Basic

```
Public Function GetAllLanguages(ByVal OrderBy As String) As LanguageData
```

Parameters

Parameters	Description
OrderBy	Valid values are language_id, language_name or any field name from the language_type table.

Returns

Array of Ektron.Cms.LanguageData

Site.GetClientDayNames Method

Returns the day name of a given content (see page 337) 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 (see page 337) 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
Keyname	The Keyname. (Required)

Returns

String

Site.GetLanguage Method

Returns the details about the given language ID.

C#

```
public LanguageData GetLanguage(int Id);
```

Visual Basic

```
Public Function GetLanguage(ByVal Id As Integer) As LanguageData
```

Parameters

Parameters	Description
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.

C#

```
public LanguageDataSet.LanguageDataTable GetLanguages(LanguageDataSet.LanguageState State,
LanguageDataSet.SortBy SortBy, 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
State	The state that the language must meet. For example, defined, active, site enabled.
SortBy	The field on which to sort the languages in the data table.
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.

C#

```
public SettingsData GetSiteVariables(int Id);
```

Visual Basic

```
Public Function GetSiteVariables(Optional ByVal Id As Integer = -1) As SettingsData
```

Parameters

Parameters	Description
Id	User 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 int GetTaskPermission(int Id, int Type);
```

Visual Basic

```
Public Function GetTaskPermission(ByVal Id As Integer, ByVal Type As Integer) As Integer
```

Parameters

Parameters	Description
Id	User or group ID
Type	User or group (1 = user, 2 = group).

Returns

Integer

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.

C#

```
public Boolean IsTaskPermissionInherited(int Id);
```

Visual Basic

```
Public Function IsTaskPermissionInherited(ByVal Id As Integer) As Boolean
```

Parameters

Parameters	Description
Id	User 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.

C#

```
public SetCookieValue(String Keyname, String CookieValue);
```

Visual Basic

```
Public Sub SetCookieValue(ByVal Keyname As String, ByVal CookieValue As String)
```

Parameters

Parameters	Description
Keyname	The Keyname. (Required)
CookieValue	Value

Site.SetTaskPermission Method

Updates the task permission(s) for the given user or group.

C#

```
public SetTaskPermission(int Id, int UserType, int PermissionType);
```

Visual Basic

```
Public Sub SetTaskPermission(ByVal Id As Integer, ByVal UserType As Integer, ByVal
PermissionType As Integer)
```

Parameters

Parameters	Description
Id	User ID or Group ID
UserType	User (1) or Group (2)

PermissionType	Ektron.Cms.Common.TaskPermission (1 = Create Task, 2 = Redirect Task, 3 = Delete Task, 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.

C#

```
public UpdateSiteEnabledLanguages(String LanguageIDs);
```

Visual Basic

```
Public Sub UpdateSiteEnabledLanguages(ByVal LanguageIDs As String)
```

Parameters

Parameters	Description
LanguageIDs	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

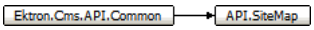
Parameters	Description
item	Set of settings data as key, value pair.

Remarks

To use this function, the user should be logged in to the system. Otherwise, the user will get the permission denied error.

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 API.SiteMap.

Members

SiteMap Methods

	Name	Description
	GetFolderBreadcrumbPath (see page 513)	Get breadcrumb path of the folder in which content (see page 337) block exists.
	GetWebsiteMap (see page 514)	Creates the site map from folder breadcrumb.
	RenderFolderBreadcrumbHtml (see page 514)	Render folder breadcrumb HTML
	RenderWebsiteMap (see page 514)	Create HTML to render on the page.

Topics

Name	Description
SiteMap Methods (see page 513)	The methods of the SiteMap class are listed here.

Legend

	Method

SiteMap Methods

	Name	Description
	GetFolderBreadcrumbPath (see page 513)	Get breadcrumb path of the folder in which content (see page 337) block exists.
	GetWebsiteMap (see page 514)	Creates the site map from folder breadcrumb.
	RenderFolderBreadcrumbHtml (see page 514)	Render folder breadcrumb HTML
	RenderWebsiteMap (see page 514)	Create HTML to render on the page.

Legend

	Method

SiteMap Methods

The methods of the SiteMap class are listed here.

Public Methods

	Name	Description
	GetFolderBreadcrumbPath (see page 513)	Get breadcrumb path of the folder in which content (see page 337) block exists.
	GetWebsiteMap (see page 514)	Creates the site map from folder breadcrumb.
	RenderFolderBreadcrumbHtml (see page 514)	Render folder breadcrumb HTML
	RenderWebsiteMap (see page 514)	Create HTML to render on the page.

Legend

	Method

SiteMap.GetFolderBreadcrumbPath Method

Get breadcrumb path of the folder in which content (see page 337) block exists.

C#

```
public Ektron.Cms.Common.SitemapPath GetFolderBreadcrumbPath(int id, Boolean isFolder);
```

Visual Basic

```
Public Function GetFolderBreadcrumbPath(ByVal id As Integer, Optional ByVal isFolder As
Boolean = False) As Ektron.Cms.Common.SitemapPath
```

Parameters

Parameters	Description
id	Content (see page 337) or Folder Id
isFolder	set it true if the id is folder id

Returns

Array of node in the path

SiteMap.GetWebsiteMap Method

Creates the site map from folder breadcrumb.

C#

```
public WebsiteMap GetWebsiteMap(int folderId, Boolean getSameLevelNodes);
```

Visual Basic

```
Public Function GetWebsiteMap(Optional ByVal folderId As Integer = 0, Optional ByVal
getSameLevelNodes As Boolean = False) As WebsiteMap
```

Parameters

Parameters	Description
folderId	Build the web sitemap (see page 512) from the folder specified.
getSameLevelNodes	True indicates include all sites at same level as folderId

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
nodes	
pathSeparator	
linkNodes	
linkTarget	
displayVerticle	

Returns

HTML

SiteMap.RenderWebsiteMap Method

Create HTML to render on the page.

C#

```
public String RenderWebsiteMap(WebsiteMap webSitemap, String displayType, String className,
int maxLevel, int startLevel, 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
webSitemap	GetWebSiteMap()
displayType	Determines how to produce HTML.
className	Stylesheet class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.
maxLevel	Set the max level of the sitemap (see page 512). Set it to 0 for unlimited.
startLevel	Set the starting level of the sitemap (see page 512). 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 API.Subscription.

Members

Subscription Methods

	Name	Description
	AddEmailFrom (see page 518)	Adds a subscription email address.
	AddSubscription (see page 518)	Adds a subscription.
	AddSubscriptionMessage (see page 518)	Adds subscription message(s)
	DeleteEmailFrom (see page 519)	Deletes a subscription email address from system.
	DeleteSubscription (see page 519)	Deletes a subscription from the system.

	DeleteSubscriptionMessage (see page 519)	Deletes the subscription message from the system.
	GetAllActiveSubscriptions (see page 519)	Loads all of the active subscriptions.
	GetAllEmailFrom (see page 520)	Returns all of the email as an array of EmailFromData.
	GetAllSubscriptionMessages (see page 520)	Loads all the subscription messages.
	GetAllSubscriptions (see page 520)	Loads all the subscriptions.
	GetFolderInheritedFrom (see page 521)	Retrieves the inherited parent ID
	GetSubscription (see page 521)	Retrieves a subscription by using its ID.
	GetSubscriptionMessage (see page 521)	Retrieves a subscription message by using the subscription ID
	GetSubscriptionMessagesForType (see page 521)	Retrieve subscription message for type.
	GetSubscriptionPropertiesForContent (see page 522)	Retrieves the subscription properties for content (see page 337) using the content (see page 337)'s ID.
	GetSubscriptionPropertiesForFolder (see page 522)	Retrieves the subscription properties for a given folder.
	GetSubscriptionsForContent (see page 522)	Loads the subscriptions for the given content (see page 337).
	GetSubscriptionsForFolder (see page 523)	Loads the subscriptions for a given folder.
	UpdateEmailFrom (see page 523)	Update the email address.
	UpdateSubscription (see page 523)	Updates the subscription.
	UpdateSubscriptionMessage (see page 524)	Updates the subscription message.

Topics

Name	Description
Subscription Methods (see page 517)	The methods of the Subscription class are listed here.

Legend

	Method

Subscription Methods

	Name	Description
	AddEmailFrom (see page 518)	Adds a subscription email address.
	AddSubscription (see page 518)	Adds a subscription.
	AddSubscriptionMessage (see page 518)	Adds subscription message(s)
	DeleteEmailFrom (see page 519)	Deletes a subscription email address from system.
	DeleteSubscription (see page 519)	Deletes a subscription from the system.
	DeleteSubscriptionMessage (see page 519)	Deletes the subscription message from the system.
	GetAllActiveSubscriptions (see page 519)	Loads all of the active subscriptions.
	GetAllEmailFrom (see page 520)	Returns all of the email as an array of EmailFromData.
	GetAllSubscriptionMessages (see page 520)	Loads all the subscription messages.
	GetAllSubscriptions (see page 520)	Loads all the subscriptions.
	GetFolderInheritedFrom (see page 521)	Retrieves the inherited parent ID

	GetSubscription (see page 521)	Retrieves a subscription by using its ID.
	GetSubscriptionMessage (see page 521)	Retrieves a subscription message by using the subscription ID
	GetSubscriptionMessagesForType (see page 521)	Retrieve subscription message for type.
	GetSubscriptionPropertiesForContent (see page 522)	Retrieves the subscription properties for content (see page 337) using the content (see page 337)'s ID.
	GetSubscriptionPropertiesForFolder (see page 522)	Retrieves the subscription properties for a given folder.
	GetSubscriptionsForContent (see page 522)	Loads the subscriptions for the given content (see page 337).
	GetSubscriptionsForFolder (see page 523)	Loads the subscriptions for a given folder.
	UpdateEmailFrom (see page 523)	Update the email address.
	UpdateSubscription (see page 523)	Updates the subscription.
	UpdateSubscriptionMessage (see page 524)	Updates the subscription message.

Legend

	Method

Subscription Methods

The methods of the Subscription class are listed here.

Public Methods

	Name	Description
	AddEmailFrom (see page 518)	Adds a subscription (see page 515) email address.
	AddSubscription (see page 518)	Adds a subscription (see page 515).
	AddSubscriptionMessage (see page 518)	Adds subscription (see page 515) message(s)
	DeleteEmailFrom (see page 519)	Deletes a subscription (see page 515) email address from system.
	DeleteSubscription (see page 519)	Deletes a subscription (see page 515) from the system.
	DeleteSubscriptionMessage (see page 519)	Deletes the subscription (see page 515) message from the system.
	GetAllActiveSubscriptions (see page 519)	Loads all of the active subscriptions.
	GetAllEmailFrom (see page 520)	Returns all of the email as an array of EmailFromData.
	GetAllSubscriptionMessages (see page 520)	Loads all the subscription (see page 515) messages.
	GetAllSubscriptions (see page 520)	Loads all the subscriptions.
	GetFolderInheritedFrom (see page 521)	Retrieves the inherited parent ID
	GetSubscription (see page 521)	Retrieves a subscription (see page 515) by using its ID.
	GetSubscriptionMessage (see page 521)	Retrieves a subscription (see page 515) message by using the subscription (see page 515) ID
	GetSubscriptionMessagesForType (see page 521)	Retrieve subscription (see page 515) message for type.
	GetSubscriptionPropertiesForContent (see page 522)	Retrieves the subscription (see page 515) properties for content (see page 337) using the content (see page 337)'s ID.
	GetSubscriptionPropertiesForFolder (see page 522)	Retrieves the subscription (see page 515) properties for a given folder.
	GetSubscriptionsForContent (see page 522)	Loads the subscriptions for the given content (see page 337).

	GetSubscriptionsForFolder (see page 523)	Loads the subscriptions for a given folder.
	UpdateEmailFrom (see page 523)	Update the email address.
	UpdateSubscription (see page 523)	Updates the subscription (see page 515).
	UpdateSubscriptionMessage (see page 524)	Updates the subscription (see page 515) message.

Legend

	Method

Subscription.AddEmailFrom Method

Adds a subscription (see page 515) email address.

C#

```
public AddEmailFrom(Collection data);
```

Visual Basic

```
Public Sub AddEmailFrom(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Email address as a collection

Subscription.AddSubscription Method

Adds a subscription (see page 515).

C#

```
public AddSubscription(Collection data);
```

Visual Basic

```
Public Sub AddSubscription(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Subscription (see page 515) name

Remarks

User login with administrator privileges are required to perform this operation.

Subscription.AddSubscriptionMessage Method

Adds subscription (see page 515) message(s)

C#

```
public AddSubscriptionMessage(Collection data);
```

Visual Basic

```
Public Sub AddSubscriptionMessage(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Subscription (see page 515) message data

Remarks

User login with administrator privileges are required to perform this operation.

Subscription.DeleteEmailFrom Method

Deletes a subscription (see page 515) email address from system.

C#

```
public DeleteEmailFrom(Collection data);
```

Visual Basic

```
Public Sub DeleteEmailFrom(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Email data

Remarks

User login with administrator privileges are required to perform this operation.

Subscription.DeleteSubscription Method

Deletes a subscription (see page 515) from the system.

C#

```
public DeleteSubscription(Collection data);
```

Visual Basic

```
Public Sub DeleteSubscription(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Subscription (see page 515) data

Remarks

User login with administrator privileges are required to perform this operation.

Subscription.DeleteSubscriptionMessage Method

Deletes the subscription (see page 515) message from the system.

C#

```
public DeleteSubscriptionMessage(Collection data);
```

Visual Basic

```
Public Sub DeleteSubscriptionMessage(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Subscription (see page 515) message data

Remarks

User login with administrator privileges are required to perform this operation.

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 (see page 515) messages.

C#

```
public EmailMessageData GetAllSubscriptionMessages();
```

Visual Basic

```
Public Function GetAllSubscriptionMessages() As EmailMessageData
```

Returns

Ektron.Cms.EmailMessageData()

Remarks

User login with administrator privileges are required to perform this operation.

Subscription.GetAllSubscriptions Method

Loads all the subscriptions.

C#

```
public SubscriptionData GetAllSubscriptions();
```

Visual Basic

```
Public Function GetAllSubscriptions() As SubscriptionData
```

Returns

Ektron.Cms.SubscriptionData()

Remarks

User login with administrator privileges are required to perform this operation.

Subscription.GetFolderInheritedFrom Method

Retrieves the inherited parent ID

C#

```
public int GetFolderInheritedFrom(int FolderId);
```

Visual Basic

```
Public Function GetFolderInheritedFrom(ByVal FolderId As Integer) As Integer
```

Parameters

Parameters	Description
FolderId	The folder's ID

Returns

Integer

Subscription.GetSubscription Method

Retrieves a subscription (see page 515) by using its ID.

C#

```
public SubscriptionData GetSubscription(int Id);
```

Visual Basic

```
Public Function GetSubscription(ByVal Id As Integer) As SubscriptionData
```

Parameters

Parameters	Description
Id	The subscription (see page 515)'s ID

Returns

Ektron.Cms.SubscriptionData

Subscription.GetSubscriptionMessage Method

Retrieves a subscription (see page 515) message by using the subscription (see page 515) ID

C#

```
public EmailMessageData GetSubscriptionMessage(int Id);
```

Visual Basic

```
Public Function GetSubscriptionMessage(ByVal Id As Integer) As EmailMessageData
```

Parameters

Parameters	Description
Id	The subscription (see page 515)'s ID

Returns

Ektron.Cms.EmailMessageData

Subscription.GetSubscriptionMessagesForType Method

Retrieve subscription (see page 515) 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
MessageType	The message type

Returns

Ektron.Cms.EmailMessageData()

Subscription.GetSubscriptionPropertiesForContent Method

Retrieves the subscription (see page 515) properties for content (see page 337) using the content (see page 337)'s ID.

C#

```
public SubscriptionPropertiesData GetSubscriptionPropertiesForContent(int ContentId);
```

Visual Basic

```
Public Function GetSubscriptionPropertiesForContent(ByVal ContentId As Integer) As
SubscriptionPropertiesData
```

Parameters

Parameters	Description
ContentId	The content (see page 337)'s ID

Returns

Ektron.Cms.SubscriptionPropertiesData

Subscription.GetSubscriptionPropertiesForFolder Method

Retrieves the subscription (see page 515) properties for a given folder.

C#

```
public SubscriptionPropertiesData GetSubscriptionPropertiesForFolder(int FolderId);
```

Visual Basic

```
Public Function GetSubscriptionPropertiesForFolder(ByVal FolderId As Integer) As
SubscriptionPropertiesData
```

Parameters

Parameters	Description
FolderId	The folder's ID

Returns

Ektron.Cms.SubscriptionPropertiesData

Subscription.GetSubscriptionsForContent Method

Loads the subscriptions for the given content (see page 337).

C#

```
public SubscriptionData GetSubscriptionsForContent(int ContentId);
```

Visual Basic

```
Public Function GetSubscriptionsForContent(ByVal ContentId As Integer) As SubscriptionData
```

Parameters

Parameters	Description
ContentId	The content (see page 337)'s ID

Returns

Ektron.Cms.SubscriptionData()

Subscription.GetSubscriptionsForFolder Method

Loads the subscriptions for a given folder.

C#

```
public SubscriptionData GetSubscriptionsForFolder(int FolderId);
```

Visual Basic

```
Public Function GetSubscriptionsForFolder(ByVal FolderId As Integer) As SubscriptionData
```

Parameters

Parameters	Description
FolderId	The folder's ID

Returns

Ektron.Cms.SubscriptionData()

Subscription.UpdateEmailFrom Method

Update the email address.

C#

```
public UpdateEmailFrom(Collection data);
```

Visual Basic

```
Public Sub UpdateEmailFrom(ByVal data As Collection)
```

Parameters

Parameters	Description
data	email data

Subscription.UpdateSubscription Method

Updates the subscription (see page 515).

C#

```
public UpdateSubscription(Collection data);
```

Visual Basic

```
Public Sub UpdateSubscription(ByVal data As Collection)
```

Parameters

Parameters	Description
data	The subscription (see page 515) details

Remarks

User login with administrator privileges are required to perform this operation.

Subscription.UpdateSubscriptionMessage Method

Updates the subscription (see page 515) message.

C#

```
public UpdateSubscriptionMessage(Collection data);
```

Visual Basic

```
Public Sub UpdateSubscriptionMessage(ByVal data As Collection)
```

Parameters

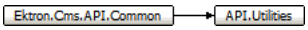
Parameters	Description
data	The subscription (see page 515) message details.

Remarks

User login with administrator privileges are required to perform this operation.

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 API.Utilities.

Members

Utilities Methods

	Name	Description
	LoadPermissions (see page 525)	Loads the permission for given object, such as content (see page 337), folder, task, etc.

Topics

Name	Description
Utilities Methods (see page 525)	The methods of the Utilities class are listed here.

Legend

	Method

Utilities Methods

	Name	Description
	LoadPermissions (see page 525)	Loads the permission for given object, such as content (see page 337), folder, task, etc.

Legend

	Method

Utilities Methods

The methods of the Utilities class are listed here.

Public Methods

	Name	Description
	LoadPermissions (see page 525)	Loads the permission for given object, such as content (see page 337), folder, task, etc.

Legend

	Method

Utilities.LoadPermissions Method

Loads the permission for given object, such as content (see page 337), folder, task, etc.

C#

```
[Obsolete("Use LoadPermissions from the Ektron.CMS.API.Permissions namespace.")]
public PermissionData LoadPermissions(int Id, String RequestType, PermissionResultType Type);
```

Visual Basic

```
<Obsolete("Use LoadPermissions from the Ektron.CMS.API.Permissions namespace.")>
Public Function LoadPermissions(ByVal Id As Integer, ByVal RequestType As String, Optional
ByVal Type As PermissionResultType = PermissionResultType.Common) As PermissionData
```

Parameters

Parameters	Description
Id	The object ID
RequestType	The object type := content (see page 337) | folder | collections | tasks
Type	The permission result type

Returns

Ektron.Cms.PermissionData

Remarks

User login with administrator privileges are required to perform this operation.

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 526)	This is namespace Ektron.Services.

Ektron.Services Namespace

This is namespace Ektron.Services.

Namespaces

Name	Description
Task (see page 526)	This is namespace Ektron.Services.Task.

Classes

	Name	Description
	ActiveDirectoryUser (see page 545)	Summary description for ActiveDirectoryUser
	Asset (see page 554)	Summary description for Asset
	Blog (see page 560)	Summary description for Blog
	BusinessRules (see page 572)	Summary description for BusinessRules
	Calendar (see page 575)	Summary description for Calendar
	CalendarEvent (see page 575)	Summary description for CalendarEvent
	CalendarEventType (see page 575)	Summary description for CalendarEventType
	Content (see page 576)	Summary description for ContentService
	ContentRating (see page 589)	Summary description for ContentRating
	CustomFields (see page 591)	Summary description for CustomFields
	Folder (see page 601)	Summary description for Folder
	Font (see page 606)	Summary description for Font
	Form (see page 609)	Summary description for Form
	Library (see page 616)	Summary description for Library
	Metadata (see page 626)	Summary description for Metadata
	Permissions (see page 630)	Summary description for Permissions
	SearchManager (see page 636)	Summary description for SearchManager
	Site (see page 640)	Summary description for Site
	SiteMap (see page 647)	Summary description for SiteMap
	ThreadedDiscussion (see page 650)	Summary description for ThreadedDiscussion
	User (see page 660)	Summary description for User

Legend

	Class

Ektron.Services.Task Namespace

This is namespace Ektron.Services.Task.

Classes

	Name	Description
	Task (see page 527)	Summary description for Task
	TaskCategory (see page 538)	Summary description for TaskCategory
	TaskCategoryType (see page 542)	Summary description for TaskCategoryType

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	Task (see page 527)	Summary description for Task
	TaskCategory (see page 538)	Summary description for TaskCategory
	TaskCategoryType (see page 542)	Summary description for TaskCategoryType

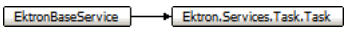
Legend

	Class

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

Members

Task Methods

	Name	Description
	AddTask (see page 529)	Adds a task in the system.
	AssignTaskToUser (see page 529)	
	AssignTaskToUsergroup (see page 530)	
	DeleteTask (see page 531)	

	DeleteTasks (see page 532)	
	GetTask (see page 533)	Get the task information.
	GetTasks (see page 534)	Get All tasks from the system.
	GetTasksAssignedByUser (see page 534)	
	GetTasksAssignedToUser (see page 535)	
	GetTasksAssignedToUsergroup (see page 536)	
	GetTaskState (see page 537)	Get the task state.
	SetStateForTask (see page 538)	Set the task state.
	UpdateTask (see page 538)	Updates the task

Topics

Name	Description
Task Methods (see page 528)	The methods of the Task class are listed here.

Legend

	Method

Task Methods

	Name	Description
	AddTask (see page 529)	Adds a task in the system.
	AssignTaskToUser (see page 529)	
	AssignTaskToUsergroup (see page 530)	
	DeleteTask (see page 531)	
	DeleteTasks (see page 532)	
	GetTask (see page 533)	Get the task information.
	GetTasks (see page 534)	Get All tasks from the system.
	GetTasksAssignedByUser (see page 534)	
	GetTasksAssignedToUser (see page 535)	
	GetTasksAssignedToUsergroup (see page 536)	
	GetTaskState (see page 537)	Get the task state.
	SetStateForTask (see page 538)	Set the task state.
	UpdateTask (see page 538)	Updates the task

Legend

	Method

Task Methods

The methods of the Task class are listed here.

Public Methods

	Name	Description
	AddTask (see page 529)	Adds a task in the system.
	AssignTaskToUser (see page 529)	
	AssignTaskToUsergroup (see page 530)	
	DeleteTask (see page 531)	
	DeleteTasks (see page 532)	

	GetTask (see page 533)	Get the task information.
	GetTasks (see page 534)	Get All tasks from the system.
	GetTasksAssignedByUser (see page 534)	
	GetTasksAssignedToUser (see page 535)	
	GetTasksAssignedToUsergroup (see page 536)	
	GetTaskState (see page 537)	Get the task state.
	SetStateForTask (see page 538)	Set the task state.
	UpdateTask (see page 538)	Updates the task

Legend

	Method

Task.AddTask Method

Adds a task in the system.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddTask", Description = "Adds a task")]
public bool AddTask(ref TaskData data);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddTask", Description = "Adds a task")>
Public Function AddTask(data As ref TaskData) As bool
```

Parameters

Parameters	Description
data	TaskData

Returns

False=if successfully added otherwise True

AssignTaskToUser Method

Overload List

	Name	Description
	Task.AssignTaskToUser (int, int) (see page 529)	Assign a task to a user.
	Task.AssignTaskToUser (int, int, int) (see page 530)	Assign a task to a user by a specified user..

Task.AssignTaskToUser Method (int, int)

Assign a task to a user.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AssignTaskToUser", Description = "Assign a task to a user.")]
public bool AssignTaskToUser(int taskId, int assignToUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AssignTaskToUser", Description = "Assign a task to a user.")>
Public Function AssignTaskToUser(taskId As Integer, assignToUserId As Integer) As bool
```

Parameters

Parameters	Description
taskId	Task ID
assignToUserId	User's ID

Returns

False=if successfully assigned to a user otherwise True

Task.AssignTaskToUser Method (int, int, int)

Assign a task to a user by a specified user..

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AssignTaskToUserByUser", Description = "Assign a task to a user.")]
public bool AssignTaskToUser(int taskId, int assignToUserId, int assignByUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AssignTaskToUserByUser", Description = "Assign a task to a user.")>
Public Function AssignTaskToUser(taskId As Integer, assignToUserId As Integer, assignByUserId
As Integer) As bool
```

Parameters

Parameters	Description
taskId	Task ID
assignToUserId	Assign to User's ID
assignByUserId	Assign by User's ID

Returns

False=if successfully assigned to a user otherwise True

AssignTaskToUsergroup Method

Overload List

	Name	Description
	Task.AssignTaskToUsergroup (int, int) (see page 530)	Assign a task to a usergroup.
	Task.AssignTaskToUsergroup (int, int, int) (see page 531)	Assign a task to a usergroup by a specified user..

Task.AssignTaskToUsergroup Method (int, int)

Assign a task to a usergroup.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AssignTaskToUsergroup", Description = "Assign a task to a
usergroup.")]
public bool AssignTaskToUsergroup(int taskId, int assignToUsergroup);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AssignTaskToUsergroup", Description = "Assign a task to a
usergroup.")>
Public Function AssignTaskToUsergroup(taskId As Integer, assignToUsergroup As Integer) As bool
```

Parameters

Parameters	Description
taskId	Task ID
assignToUsergroup	Assign to Usergroup's ID

Returns

False=if successfully assigned to a usergroup otherwise True

Task.AssignTaskToUsergroup Method (int, int, int)

Assign a task to a usergroup by a specified user..

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AssignTaskToUsergroupByUser", Description = "Assign a task to a
usergroup by a specified user.")]
public bool AssignTaskToUsergroup(int taskId, int assignToUsergroup, int assignByUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AssignTaskToUsergroupByUser", Description = "Assign a task to a
usergroup by a specified user.")>
Public Function AssignTaskToUsergroup(taskId As Integer, assignToUsergroup As Integer,
assignByUserId As Integer) As bool
```

Parameters

Parameters	Description
taskId	Task ID
assignToUsergroup	Assign to Usergroup's ID
assignByUserId	Assign by User's ID

Returns

False=if successfully assigned to a usergroup otherwise True

DeleteTask Method

Overload List

	Name	Description
	Task.DeleteTask (int) (see page 531)	Mark the task deleted.
	Task.DeleteTask (int, bool) (see page 532)	Mark the task deleted or delete the task permanently.

Task.DeleteTask Method (int)

Mark the task deleted.

C#

```
[SoapHeader("AuthHeader")]
```

```
[WebMethod(MessageName = "DeleteTask", Description = "Mark the task deleted.")]
public bool DeleteTask(int taskId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteTask", Description = "Mark the task deleted.")>
Public Function DeleteTask(taskId As Integer) As bool
```

Parameters

Parameters	Description
taskId	Task ID

Returns

False=if successfully deleted otherwise True

Task.DeleteTask Method (int, bool)

Mark the task deleted or delete the task permanently.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteTaskPermanently", Description = "Mark the task deleted or
delete the task permanently.")]
public bool DeleteTask(int taskId, bool permanently);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteTaskPermanently", Description = "Mark the task deleted or
delete the task permanently.")>
Public Function DeleteTask(taskId As Integer, permanently As bool) As bool
```

Parameters

Parameters	Description
taskId	Task ID
Permanently	True=Delete the task

Returns

False=if successfully deleted otherwise True

DeleteTasks Method

Overload List

	Name	Description
	Task.DeleteTasks (int[]) (see page 532)	Mark tasks deleted.
	Task.DeleteTasks (int[], bool) (see page 533)	Mark tasks deleted or delete tasks permanently.

Task.DeleteTasks Method (int[])

Mark tasks deleted.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteTasks", Description = "Mark tasks deleted.")]
public bool DeleteTasks(int[] taskId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteTasks", Description = "Mark tasks deleted.")>
Public Function DeleteTasks(taskId As int[]) As bool
```

Parameters

Parameters	Description
taskId	Array of Task 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 (int[], bool)

Mark tasks deleted or delete tasks permanently.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteTasksPermanently", Description = "Mark tasks deleted or delete
tasks permanently.")]
public bool DeleteTasks(int[] taskId, bool permanently);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteTasksPermanently", Description = "Mark tasks deleted or delete
tasks permanently.")>
Public Function DeleteTasks(taskId As int[], permanently As bool) As bool
```

Parameters

Parameters	Description
taskId	Array of Task 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")]
[WebMethod(MessageName = "GetTask", Description = "Get the task information.")]
public TaskData GetTask(int taskId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTask", Description = "Get the task information.")>
Public Function GetTask(taskId As Integer) As TaskData
```

Parameters

Parameters	Description
taskId	Task Id

Returns

Task Data

Task.GetTasks Method

Get All tasks from the system.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetTasks", Description = "")]
public TaskData[] GetTasks();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTasks", Description = "")>
Public Function GetTasks() As TaskData[]
```

Returns

Array of task data.

GetTasksAssignedByUser Method

Overload List

	Name	Description
	Task.GetTasksAssignedByUser (int) (see page 534)	Get all tasks assigned by the user.
	Task.GetTasksAssignedByUser (int, Ektron.Cms.Common.EkEnumeration.TaskState) (see page 535)	Get all task assign by the user by a state of tasks specified.

Task.GetTasksAssignedByUser Method (int)

Get all tasks assigned by the user.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetTasksAssignedByUser", Description = "Get all tasks assigned by
the user.")]
public TaskData[] GetTasksAssignedByUser(int userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTasksAssignedByUser", Description = "Get all tasks assigned by
the user.")>
Public Function GetTasksAssignedByUser(userId As Integer) As TaskData[]
```

Parameters

Parameters	Description
userId	User's ID

Returns

Array of tasks.

Task.GetTasksAssignedByUser Method (int, Ektron.Cms.Common.EkEnumeration.TaskState)

Get all task assign by the user by a state of tasks specified.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetTasksAssignedByUserByState", Description = "")]
public TaskData[] GetTasksAssignedByUser(int userId, Ektron.Cms.Common.EkEnumeration.TaskState
taskState);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTasksAssignedByUserByState", Description = "")>
Public Function GetTasksAssignedByUser(userId As Integer, taskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As TaskData[]
```

Parameters

Parameters	Description
userId	User's ID
taskState	Task State

Returns

Array of tasks.

GetTasksAssignedToUser Method

Overload List

	Name	Description
	Task.GetTasksAssignedToUser (int) (see page 535)	Get all tasks assigned to the user.
	Task.GetTasksAssignedToUser (int, Ektron.Cms.Common.EkEnumeration.TaskState) (see page 536)	Get all tasks assigned to the user by a state of the task specified.

Task.GetTasksAssignedToUser Method (int)

Get all tasks assigned to the user.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUser", Description = "Get all tasks assigned to
the user.")]
public TaskData[] GetTasksAssignedToUser(int userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUser", Description = "Get all tasks assigned to
the user.")>
Public Function GetTasksAssignedToUser(userId As Integer) As TaskData[]
```

Parameters

Parameters	Description
userId	User's ID

Returns

Array of tasks.

Task.GetTasksAssignedToUser Method (int, Ektron.Cms.Common.EkEnumeration.TaskState)

Get all tasks assigned to the user by a state of the task specified.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUserByState", Description = "Get all tasks
assigned to the user by a state of the task specified.")]
public TaskData[] GetTasksAssignedToUser(int userId, Ektron.Cms.Common.EkEnumeration.TaskState
taskState);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUserByState", Description = "Get all tasks
assigned to the user by a state of the task specified.")>
Public Function GetTasksAssignedToUser(userId As Integer, taskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As TaskData[]
```

Parameters

Parameters	Description
userId	User's ID
taskState	Task State

Returns

Array of tasks.

GetTasksAssignedToUsergroup Method

Overload List

	Name	Description
	Task.GetTasksAssignedToUsergroup (int) (see page 536)	Get all tasks assigned to the usergroup.
	Task.GetTasksAssignedToUsergroup (int, Ektron.Cms.Common.EkEnumeration.TaskState) (see page 537)	Get all tasks assigned to the usergroup by a state of tasks specified.

Task.GetTasksAssignedToUsergroup Method (int)

Get all tasks assigned to the usergroup.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUsergroup", Description = "Get all tasks assigned
to the usergroup.")]
public TaskData[] GetTasksAssignedToUsergroup(int groupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUsergroup", Description = "Get all tasks assigned
to the usergroup.")>
Public Function GetTasksAssignedToUsergroup(groupId As Integer) As TaskData[]
```

Parameters

Parameters	Description
groupId	Usergroup's ID

Returns

Array of tasks.

Task.GetTasksAssignedToUsergroup Method (int, Ektron.Cms.Common.EkEnumeration.TaskState)

Get all tasks assigned to the usergroup by a state of tasks specified.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUsergroupByState", Description = "Get all tasks
assigned to the usergroup by a state of tasks specified.")]
public TaskData[] GetTasksAssignedToUsergroup(int groupId,
Ektron.Cms.Common.EkEnumeration.TaskState taskState);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUsergroupByState", Description = "Get all tasks
assigned to the usergroup by a state of tasks specified.")>
Public Function GetTasksAssignedToUsergroup(groupId As Integer, taskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As TaskData[]
```

Parameters

Parameters	Description
groupId	Usergroup's ID
taskState	Task State

Returns

Array of tasks.

Task.GetTaskState Method

Get the task state.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetTaskState", Description = "Get the task state.")]
public Ektron.Cms.Common.EkEnumeration.TaskState GetTaskState(int taskId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTaskState", Description = "Get the task state.")>
Public Function GetTaskState(taskId As Integer) As Ektron.Cms.Common.EkEnumeration.TaskState
```

Parameters

Parameters	Description
taskId	Task ID

Returns

If successful returns task state otherwise it will throw an error.

Task.SetStateForTask Method

Set the task state.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "SetStateForTask", Description = "Set the task state.")]
public bool SetStateForTask(int taskId, Ektron.Cms.Common.EkEnumeration.TaskState
newTaskState);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "SetStateForTask", Description = "Set the task state.")>
Public Function SetStateForTask(taskId As Integer, newTaskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As bool
```

Parameters

Parameters	Description
taskId	Task ID
newTaskState	New Task State

Returns

False=if successfully state of the changed otherwise True

Task.UpdateTask Method

Updates the task

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateTask", Description = "Updates the task")]
public bool UpdateTask(ref TaskData data);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UpdateTask", Description = "Updates the task")>
Public Function UpdateTask(data As ref TaskData) As bool
```

Parameters

Parameters	Description
data	Task Data

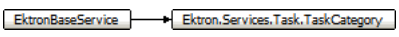
Returns

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

Members

TaskCategory Methods

	Name	Description
	AddTaskCategory (see page 540)	Adds a task category.
	DeleteTaskCategory (see page 540)	Deletes the task category.
	GetAllCategories (see page 540)	Gets All Task categories and task types belongs to those categories.
	GetTaskCategory (see page 541)	Gets the specified task category and task types belongs to the category.
	UpdateTaskCategory (see page 541)	Updates the category title.

Topics

Name	Description
TaskCategory Methods (see page 539)	The methods of the TaskCategory class are listed here.

Legend

	Method

TaskCategory Methods

	Name	Description
	AddTaskCategory (see page 540)	Adds a task category.
	DeleteTaskCategory (see page 540)	Deletes the task category.
	GetAllCategories (see page 540)	Gets All Task categories and task types belongs to those categories.
	GetTaskCategory (see page 541)	Gets the specified task category and task types belongs to the category.
	UpdateTaskCategory (see page 541)	Updates the category title.

Legend

	Method

TaskCategory Methods

The methods of the TaskCategory class are listed here.

Public Methods

	Name	Description
	AddTaskCategory (see page 540)	Adds a task category.

	DeleteTaskCategory (see page 540)	Deletes the task category.
	GetAllCategories (see page 540)	Gets All Task categories and task types belongs to those categories.
	GetTaskCategory (see page 541)	Gets the specified task category and task types belongs to the category.
	UpdateTaskCategory (see page 541)	Updates the category title.

Legend

	Method

TaskCategory.AddTaskCategory Method

Adds a task category.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddTaskCategory", Description = "Adds a task category.")]
public int AddTaskCategory(string title);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddTaskCategory", Description = "Adds a task category.")>
Public Function AddTaskCategory(title As string) As Integer
```

Parameters

Parameters	Description
title	Category Title

Returns

Category Id

TaskCategory.DeleteTaskCategory Method

Deletes the task category.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteTaskCategory", Description = "Deletes the task category.")]
public bool DeleteTaskCategory(int categoryId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteTaskCategory", Description = "Deletes the task category.")>
Public Function DeleteTaskCategory(categoryId As Integer) As bool
```

Parameters

Parameters	Description
categoryId	category id

Returns

False=if successfully deleted otherwise True

TaskCategory.GetAllCategories Method

Gets All Task categories and task types belongs to those categories.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllCategories", Description = "Gets All Task categories and task
types belongs to those categories.")]
public TaskCategoryData[] GetAllCategories();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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")]
[WebMethod(MessageName = "GetTaskCategory", Description = "Gets the specified task category
and task types belongs to the category.")]
public TaskCategoryData GetTaskCategory(int categoryID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTaskCategory", Description = "Gets the specified task category
and task types belongs to the category.")>
Public Function GetTaskCategory(categoryID As Integer) As TaskCategoryData
```

Parameters

Parameters	Description
categoryID	category id

Returns

TaskCategoryData

TaskCategory.UpdateTaskCategory Method

Updates the category title.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateTaskCategory", Description = "Updates the category title.")]
public bool UpdateTaskCategory(int categoryId, string title);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UpdateTaskCategory", Description = "Updates the category title.")>
Public Function UpdateTaskCategory(categoryId As Integer, title As string) As bool
```

Parameters

Parameters	Description
categoryId	Category Id
title	New Title

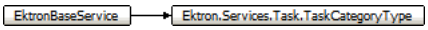
Returns

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

Members

TaskCategoryType Methods

	Name	Description
	AddTaskType (see page 543)	Adds a task type to a category.
	DeleteTaskType (see page 543)	Deletes the task type.
	GetTaskType (see page 544)	Gets the task type information.
	UpdateTaskType (see page 544)	Updates the task type.

Topics

Name	Description
TaskCategoryType Methods (see page 543)	The methods of the TaskCategoryType class are listed here.

Legend

	Method

TaskCategoryType Methods

	Name	Description
	AddTaskType (see page 543)	Adds a task type to a category.
	DeleteTaskType (see page 543)	Deletes the task type.
	GetTaskType (see page 544)	Gets the task type information.
	UpdateTaskType (see page 544)	Updates the task type.

Legend

	Method

TaskCategoryType Methods

The methods of the TaskCategoryType class are listed here.

Public Methods

	Name	Description
	AddTaskType (see page 543)	Adds a task type to a category.
	DeleteTaskType (see page 543)	Deletes the task type.
	GetTaskType (see page 544)	Gets the task type information.
	UpdateTaskType (see page 544)	Updates the task type.

Legend

	Method

TaskCategoryType.AddTaskType Method

Adds a task type to a category.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddTaskType", Description = "Adds a task type to a category.")]
public int AddTaskType(string title, string description, int categoryId, bool active);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddTaskType", Description = "Adds a task type to a category.")>
Public Function AddTaskType(title As string, description As string, categoryId As Integer,
active As bool) As Integer
```

Parameters

Parameters	Description
title	Title
description	Description
categoryId	Task Category ID
active	Set as active

Returns

Task Type ID

TaskCategoryType.DeleteTaskType Method

Deletes the task type.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteTaskType", Description = "Deletes the task type.")]
public bool DeleteTaskType(int taskTypeId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteTaskType", Description = "Deletes the task type.")>
Public Function DeleteTaskType(taskTypeId As Integer) As bool
```

Parameters

Parameters	Description
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")]
[WebMethod(MessageName = "GetTaskType", Description = "Gets a task type")]
public TaskTypeData GetTaskType(int taskTypeId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTaskType", Description = "Gets a task type")>
Public Function GetTaskType(taskTypeId As Integer) As TaskTypeData
```

Parameters

Parameters	Description
taskTypeId	TaskTypeId

Returns

TaskTypeData

TaskCategoryType.UpdateTaskType Method

Updates the task type.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateTaskType", Description = "Updates the task type.")]
public bool UpdateTaskType(int taskTypeId, string title, string description, bool active);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UpdateTaskType", Description = "Updates the task type.")>
Public Function UpdateTaskType(taskTypeId As Integer, title As string, description As string,
active As bool) As bool
```

Parameters

Parameters	Description
taskTypeId	Task Type Id
title	Title
description	Description
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 545)	Summary description for ActiveDirectoryUser
	Asset (see page 554)	Summary description for Asset
	Blog (see page 560)	Summary description for Blog
	BusinessRules (see page 572)	Summary description for BusinessRules
	Calendar (see page 575)	Summary description for Calendar
	CalendarEvent (see page 575)	Summary description for CalendarEvent
	CalendarEventType (see page 575)	Summary description for CalendarEventType
	Content (see page 576)	Summary description for ContentService
	ContentRating (see page 589)	Summary description for ContentRating
	CustomFields (see page 591)	Summary description for CustomFields
	Folder (see page 601)	Summary description for Folder
	Font (see page 606)	Summary description for Font
	Form (see page 609)	Summary description for Form
	Library (see page 616)	Summary description for Library
	Metadata (see page 626)	Summary description for Metadata
	Permissions (see page 630)	Summary description for Permissions
	SearchManager (see page 636)	Summary description for SearchManager
	Site (see page 640)	Summary description for Site
	SiteMap (see page 647)	Summary description for SiteMap
	ThreadedDiscussion (see page 650)	Summary description for ThreadedDiscussion
	User (see page 660)	Summary description for User

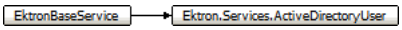
Legend

	Class

ActiveDirectoryUser Class

Summary description for ActiveDirectoryUser

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

Members

ActiveDirectoryUser Methods

	Name	Description
	AddADGroupToCMS (see page 548)	This function allows you add Active Directory groups to the CMS.
	DeSynchUserGroups (see page 548)	Updates the group with a new group name.
	DeSynchUsers (see page 548)	Update the user with new user details.
	GetADMapping (see page 549)	Loads the CMS and Active Directory name.
	GetADStatus (see page 549)	Used to load required Synch, Desynch and relationship flags of the configured Active Directory.
	GetAvailableADGroups (see page 550)	Loads all of the Active Directory groups by a given domain and filter.
	GetAvailableADUsers (see page 550)	Loads all available Active Directory users.
	GetCMSGroupsToSync (see page 551)	Loads all the CMS groups that need to be synchronized.
	GetCMSRelationshipsToSync (see page 551)	Loads all the CMS relationships that need to be synchronized with Active Directory.
	GetCMSUsersToSync (see page 551)	Loads all the CMS users that need to be synchronized with Active Directory.
	GetDomains (see page 552)	Loads a list of domains from Active Directory.
	MapCMSUserGroupToAD (see page 552)	Updates the given CMS group corresponding to the Active Directory group.
	MapCMSUserToAD (see page 553)	Maps the given user ID to the corresponding Active Directory entry.
	UpdateADMapping (see page 553)	Updates the Active directory information into the CMS based on the user ID and user data.
	UpdateADUsersGroups (see page 553)	Updates Active Directory users in a CMS group.
	UpdateUsersInfoFromAD (see page 554)	Updates the user details from the Active Directory details.

Topics

Name	Description
ActiveDirectoryUser Methods (see page 547)	The methods of the ActiveDirectoryUser class are listed here.

Legend

	Method

ActiveDirectoryUser Methods

	Name	Description
	AddADGroupToCMS (see page 548)	This function allows you add Active Directory groups to the CMS.
	DeSynchUserGroups (see page 548)	Updates the group with a new group name.
	DeSynchUsers (see page 548)	Update the user with new user details.
	GetADMapping (see page 549)	Loads the CMS and Active Directory name.
	GetADStatus (see page 549)	Used to load required Synch, Desynch and relationship flags of the configured Active Directory.
	GetAvailableADGroups (see page 550)	Loads all of the Active Directory groups by a given domain and filter.
	GetAvailableADUsers (see page 550)	Loads all available Active Directory users.

	GetCMSGroupsToSync (see page 551)	Loads all the CMS groups that need to be synchronized.
	GetCMSRelationshipsToSync (see page 551)	Loads all the CMS relationships that need to be synchronized with Active Directory.
	GetCMSUsersToSync (see page 551)	Loads all the CMS users that need to be synchronized with Active Directory.
	GetDomains (see page 552)	Loads a list of domains from Active Directory.
	MapCMSUserGroupToAD (see page 552)	Updates the given CMS group corresponding to the Active Directory group.
	MapCMSUserToAD (see page 553)	Maps the given user ID to the corresponding Active Directory entry.
	UpdateADMapping (see page 553)	Updates the Active directory information into the CMS based on the user ID and user data.
	UpdateADUsersGroups (see page 553)	Updates Active Directory users in a CMS group.
	UpdateUsersInfoFromAD (see page 554)	Updates the user details from the Active Directory details.

Legend

	Method

ActiveDirectoryUser Methods

The methods of the ActiveDirectoryUser class are listed here.

Public Methods

	Name	Description
	AddADGroupToCMS (see page 548)	This function allows you add Active Directory groups to the CMS.
	DeSynchUserGroups (see page 548)	Updates the group with a new group name.
	DeSynchUsers (see page 548)	Update the user with new user details.
	GetADMapping (see page 549)	Loads the CMS and Active Directory name.
	GetADStatus (see page 549)	Used to load required Synch, Desynch and relationship flags of the configured Active Directory.
	GetAvailableADGroups (see page 550)	Loads all of the Active Directory groups by a given domain and filter.
	GetAvailableADUsers (see page 550)	Loads all available Active Directory users.
	GetCMSGroupsToSync (see page 551)	Loads all the CMS groups that need to be synchronized.
	GetCMSRelationshipsToSync (see page 551)	Loads all the CMS relationships that need to be synchronized with Active Directory.
	GetCMSUsersToSync (see page 551)	Loads all the CMS users that need to be synchronized with Active Directory.
	GetDomains (see page 552)	Loads a list of domains from Active Directory.
	MapCMSUserGroupToAD (see page 552)	Updates the given CMS group corresponding to the Active Directory group.
	MapCMSUserToAD (see page 553)	Maps the given user ID to the corresponding Active Directory entry.
	UpdateADMapping (see page 553)	Updates the Active directory information into the CMS based on the user ID and user data.
	UpdateADUsersGroups (see page 553)	Updates Active Directory users in a CMS group.
	UpdateUsersInfoFromAD (see page 554)	Updates the user details from the Active Directory details.

Legend

	Method

ActiveDirectoryUser.AddADGroupToCMS Method

This function allows you add Active Directory groups to the CMS.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddADGroupToCMS", Description = "")]
public void AddADGroupToCMS(object GroupPaths);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddADGroupToCMS", Description = "")>
Public Function AddADGroupToCMS(GroupPaths As object) As void
```

Parameters

Parameters	Description
GroupPaths	A collection of Active Directory 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeSynchUserGroups", Description = "")]
public AdDeSyncGroupData[] DeSynchUserGroups(bool Flag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeSynchUserGroups", Description = "")>
Public Function DeSynchUserGroups(Flag As bool) As AdDeSyncGroupData[]
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeSynchUsers", Description = "")]
public AdDeSyncUserData[] DeSynchUsers(bool Flag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeSynchUsers", Description = "")>
Public Function DeSynchUsers(Flag As bool) As AdDeSyncUserData[]
```

Parameters

Parameters	Description
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 name.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetADMapping", Description = "")]
public AdMappingData[] GetADMapping(int CallerId, string MapType, int Map, int Exclude, int
Editable);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetADMapping", Description = "")>
Public Function GetADMapping(CallerId As Integer, MapType As string, Map As Integer, Exclude
As Integer, Editable As Integer) As AdMappingData[]
```

Parameters

Parameters	Description
CallerId	Who is calling this function.
MapType	Possible value is “userprop” (user properties).
Map	Ektron.Cms.Common.EkEnumeration.MapTypes (include, exclude and dontconsider are the possible values).
Exclude	Flag (0 – false, 1-true).
Editable	Flag(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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetADStatus", Description = "")]
public AdSyncData GetADStatus();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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 groups by a given domain and filter.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAvailableADGroups", Description = "")]
public GroupData[] GetAvailableADGroups(string Filter, string Domain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAvailableADGroups", Description = "")>
Public Function GetAvailableADGroups(Filter As string, Domain As string) As GroupData[]
```

Parameters

Parameters	Description
Filter	Filter by complete or a specific group name.
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 users.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAvailableADUsers", Description = "")]
public UserData[] GetAvailableADUsers(object Attributes, object Filter, string OrderBy, string
Domain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAvailableADUsers", Description = "")>
Public Function GetAvailableADUsers(Attributes As object, Filter As object, OrderBy As string,
Domain As string) As UserData[]
```

Parameters

Parameters	Description
Attributes	Possible attributes are firstname, lastname, username or any(*).
Filter	Possible filters are firstname, lastname, username or any(*).
OrderBy	Any specific order such as username, firstname, lastname and domain.
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetCMSGroupsToSync", Description = "")]
public GroupData[] GetCMSGroupsToSync(int MaxReturn);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetCMSGroupsToSync", Description = "")>
Public Function GetCMSGroupsToSync(MaxReturn As Integer) As GroupData[]
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetCMSRelationshipsToSync", Description = "")]
public UserGroupData[] GetCMSRelationshipsToSync(int MaxReturn);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetCMSRelationshipsToSync", Description = "")>
Public Function GetCMSRelationshipsToSync(MaxReturn As Integer) As UserGroupData[]
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetCMSUsersToSync", Description = "")]
public UserData[] GetCMSUsersToSync(int MaxReturn);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetCMSUsersToSync", Description = "")>
Public Function GetCMSUsersToSync(MaxReturn As Integer) As UserData[]
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetDomains", Description = "")]
public DomainData[] GetDomains(int ShowAll, int CheckIntegration);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetDomains", Description = "")>
Public Function GetDomains(ShowAll As Integer, CheckIntegration As Integer) As DomainData[]
```

Parameters

Parameters	Description
ShowAll	flag(1 = show all).
CheckIntegration	Check whether CMS is configured to integrate Active Directory (1 = check otherwise resumed).

Returns

Ektron.Cms.DomainData()

ActiveDirectoryUser.MapCMSUserGroupToAD Method

Updates the given CMS group corresponding to the Active Directory group.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "MapCMSUserGroupToAD", Description = "")]
public void MapCMSUserGroupToAD(int GroupId, string GroupName, string Domain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "MapCMSUserGroupToAD", Description = "")>
Public Function MapCMSUserGroupToAD(GroupId As Integer, GroupName As string, Domain As string)
As void
```

Parameters

Parameters	Description
GroupId	CMS group ID.
GroupName	CMS group name.
Domain	The Active Directory 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 entry.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "MapCMSUserToAD", Description = "")]
public void MapCMSUserToAD(int UserId, string UserName, string Domain, int PullInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "MapCMSUserToAD", Description = "")>
Public Function MapCMSUserToAD(UserId As Integer, UserName As string, Domain As string,
PullInfo As Integer) As void
```

Parameters

Parameters	Description
UserId	The user's ID.
UserName	The user's name.
Domain	The user's domain.
PullInfo	Flag to load Active Directory information. (1 = Load Active Directory 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateADMapping", Description = "")]
public void UpdateADMapping(int UserId, object pagedata);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UpdateADMapping", Description = "")>
Public Function UpdateADMapping(UserId As Integer, pagedata As object) As void
```

Parameters

Parameters	Description
UserId	The user's ID.
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 users in a CMS group.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateADUsersGroups", Description = "")]
```

```
public void UpdateADUsersGroups(string UserName, string Domain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UpdateADUsersGroups", Description = "")>
Public Function UpdateADUsersGroups(UserName As string, Domain As string) As void
```

Parameters

Parameters	Description
UserName	The user's username.
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 details.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateUsersInfoFromAD", Description = "")]
public void UpdateUsersInfoFromAD(string UserName, string Domain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UpdateUsersInfoFromAD", Description = "")>
Public Function UpdateUsersInfoFromAD(UserName As string, Domain As string) As void
```

Parameters

Parameters	Description
UserName	The user's username.
Domain	The user's Domain.

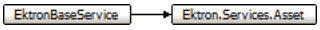
Remarks

A user with administrator privileges must be logged in to perform this operation.

Asset Class

Summary description for Asset

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

Members

Asset Methods

	Name	Description
	GetAssetMgtConfigInfo (see page 556)	Loads all of the registered plug-in configuration details.
	GetAssetMgtInfo (see page 556)	Loads all of the registered plug-in type details.
	GetAssetSupertypes (see page 557)	Loads all of the asset supertypes supported by documents add-in.
	GetCreateDragDropSnippet (see page 557)	Loads the drag and drop client snippet used with the Document Management functionality.
	GetCreateEktronExplorerSnippet (see page 557)	Loads the explorer snippet used with the Document Management functionality.
	GetCreateSnippet (see page 558)	Loads the create document snippet used with the Document Management functionality.
	GetDragDropWOSnippet (see page 558)	Loads the edit snippet for the given document when working offline. Used with the Document Management functionality.
	GetEditSnippet (see page 558)	Loads the edit snippet for the given document. Used with the Document Management functionality.
	GetMyAssetMgtID (see page 559)	Retrieves the Document Management functionality client ID.
	MakeCopyDMSAsset (see page 559)	Create a copy of DMS Asset.
	SetAssetMgtConfigInfo (see page 560)	Updates the Document Management configuration information.

Topics

Name	Description
Asset Methods (see page 556)	The methods of the Asset class are listed here.

Legend

	Method

Asset Methods

	Name	Description
	GetAssetMgtConfigInfo (see page 556)	Loads all of the registered plug-in configuration details.
	GetAssetMgtInfo (see page 556)	Loads all of the registered plug-in type details.
	GetAssetSupertypes (see page 557)	Loads all of the asset supertypes supported by documents add-in.
	GetCreateDragDropSnippet (see page 557)	Loads the drag and drop client snippet used with the Document Management functionality.
	GetCreateEktronExplorerSnippet (see page 557)	Loads the explorer snippet used with the Document Management functionality.
	GetCreateSnippet (see page 558)	Loads the create document snippet used with the Document Management functionality.
	GetDragDropWOSnippet (see page 558)	Loads the edit snippet for the given document when working offline. Used with the Document Management functionality.
	GetEditSnippet (see page 558)	Loads the edit snippet for the given document. Used with the Document Management functionality.
	GetMyAssetMgtID (see page 559)	Retrieves the Document Management functionality client ID.
	MakeCopyDMSAsset (see page 559)	Create a copy of DMS Asset.
	SetAssetMgtConfigInfo (see page 560)	Updates the Document Management configuration information.

Legend

	Method

Asset Methods

The methods of the Asset class are listed here.

Public Methods

	Name	Description
	GetAssetMgtConfigInfo (see page 556)	Loads all of the registered plug-in configuration details.
	GetAssetMgtInfo (see page 556)	Loads all of the registered plug-in type details.
	GetAssetSupertypes (see page 557)	Loads all of the asset supertypes supported by documents add-in.
	GetCreateDragDropSnippet (see page 557)	Loads the drag and drop client snippet used with the Document Management functionality.
	GetCreateEktronExplorerSnippet (see page 557)	Loads the explorer snippet used with the Document Management functionality.
	GetCreateSnippet (see page 558)	Loads the create document snippet used with the Document Management functionality.
	GetDragDropWOSnippet (see page 558)	Loads the edit snippet for the given document when working offline. Used with the Document Management functionality.
	GetEditSnippet (see page 558)	Loads the edit snippet for the given document. Used with the Document Management functionality.
	GetMyAssetMgtID (see page 559)	Retrieves the Document Management functionality client ID.
	MakeCopyDMSAsset (see page 559)	Create a copy of DMS Asset.
	SetAssetMgtConfigInfo (see page 560)	Updates the Document Management configuration information.

Legend

	Method

Asset.GetAssetMgtConfigInfo Method

Loads all of the registered plug-in configuration details.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAssetMgtConfigInfo", Description = "")]
public AssetConfigInfo[] GetAssetMgtConfigInfo();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAssetMgtInfo", Description = "")]
public AssetInfoData[] GetAssetMgtInfo();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
```

```
<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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAssetSupertypes", Description = "")]
public AssetInfoData[] GetAssetSupertypes();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAssetSupertypes", Description = "")>
Public Function GetAssetSupertypes() As AssetInfoData[]
```

Returns

Ektron.Cms.AssetInfoData()

Asset.GetCreateDragDropSnippet Method

Loads the drag and drop client snippet used with the Document Management functionality.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetCreateDragDropSnippet", Description = "")]
public string GetCreateDragDropSnippet();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetCreateDragDropSnippet", Description = "")>
Public Function GetCreateDragDropSnippet() As string
```

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetCreateEktronExplorerSnippet Method

Loads the explorer snippet used with the Document Management functionality.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetCreateEktronExplorerSnippet", Description = "")]
public string GetCreateEktronExplorerSnippet();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetCreateEktronExplorerSnippet", Description = "")>
Public Function GetCreateEktronExplorerSnippet() As string
```

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetCreateSnippet Method

Loads the create document snippet used with the Document Management functionality.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetCreateSnippet", Description = "")]
public string GetCreateSnippet(int ContentType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetCreateSnippet", Description = "")>
Public Function GetCreateSnippet(ContentType As Integer) As string
```

Parameters

Parameters	Description
ContentType	The type of the content (see page 337), such as office documents or managed files.

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetDragDropWOSnippet Method

Loads the edit snippet for the given document when working offline. Used with the Document Management functionality.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetDragDropWOSnippet", Description = "")]
public string GetDragDropWOSnippet(string AssetId, int ContentType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetDragDropWOSnippet", Description = "")>
Public Function GetDragDropWOSnippet(AssetId As string, ContentType As Integer) As string
```

Parameters

Parameters	Description
AssetId	The document's ID (Required)
ContentType	The type of the content (see page 337), 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.GetEditSnippet Method

Loads the edit snippet for the given document. Used with the Document Management functionality.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetEditSnippet", Description = "")]
public string GetEditSnippet(string AssetId, int ContentType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetEditSnippet", Description = "")>
Public Function GetEditSnippet(AssetId As string, ContentType As Integer) As string
```

Parameters

Parameters	Description
AssetId	The document's ID (Required)
ContentType	The type of the content (see page 337), 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetMyAssetMgtID", Description = "")]
public string GetMyAssetMgtID();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetMyAssetMgtID", Description = "")>
Public Function GetMyAssetMgtID() As string
```

Returns

String

Asset.MakeCopyDMSAsset Method

Create a copy of DMS Asset.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "MakeCopyDMSAsset", Description = "")]
public bool MakeCopyDMSAsset(string AssetID, string AssetVersion);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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 login with create content (see page 337) privileges are required to perform this operation.

Asset.SetAssetMgtConfigInfo Method

Updates the Document Management configuration information.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "SetAssetMgtConfigInfo", Description = "")]
public bool SetAssetMgtConfigInfo(AssetConfigInfo[] AssetConfig);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "SetAssetMgtConfigInfo", Description = "")>
Public Function SetAssetMgtConfigInfo(AssetConfig As AssetConfigInfo[]) As bool
```

Parameters

Parameters	Description
asset_config	The Document Management configuration data

Returns

Boolean

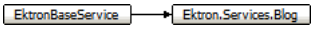
Remarks

User login with administrator privileges are required to perform this operation.

Blog Class

Summary description for Blog

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

Members

Blog Methods

	Name	Description
	AddBlog (see page 562)	Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.

	AddCommentForPost (see page 563)	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 563)	Adds a post to a given blog. Accepts a ContentData class as the post content (see page 337). Return new post ID.
	DeleteBlog (see page 564)	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.
	DeletePost (see page 564)	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 564)	Returns a blank BlogPostData class. Used Primarily in the Workarea.
	GetBlog (see page 565)	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.
	GetBlogData (see page 566)	Overloaded Method to obtain a BlogData Class.
	GetBlogDataForPost (see page 570)	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 570)	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 570)	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.
	GetPost (see page 572)	Gets a blog post for a given id. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData Class.

Topics

Name	Description
Blog Methods (see page 562)	The methods of the Blog class are listed here.

Legend

	Method

Blog Methods

	Name	Description
	AddBlog (see page 562)	Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.
	AddCommentForPost (see page 563)	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 563)	Adds a post to a given blog. Accepts a ContentData class as the post content (see page 337). Return new post ID.
	DeleteBlog (see page 564)	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.
	DeletePost (see page 564)	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 564)	Returns a blank BlogPostData class. Used Primarily in the Workarea.
	GetBlog (see page 565)	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.
	GetBlogData (see page 566)	Overloaded Method to obtain a BlogData Class.
	GetBlogDataForPost (see page 570)	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 570)	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 570)	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.
	GetPost (see page 572)	Gets a blog post for a given id. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData Class.

Legend

	Method

Blog Methods

The methods of the Blog class are listed here.

Public Methods

	Name	Description
	AddBlog (see page 562)	Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.
	AddCommentForPost (see page 563)	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 563)	Adds a post to a given blog. Accepts a ContentData class as the post content (see page 337). Return new post ID.
	DeleteBlog (see page 564)	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.
	DeletePost (see page 564)	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 564)	Returns a blank BlogPostData class. Used Primarily in the Workarea.
	GetBlog (see page 565)	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.
	GetBlogData (see page 566)	Overloaded Method to obtain a BlogData Class.
	GetBlogDataForPost (see page 570)	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 570)	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 570)	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.
	GetPost (see page 572)	Gets a blog post for a given id. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData Class.

Legend

	Method

Blog.AddBlog Method

Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddBlog", Description = "")]
public int AddBlog(int 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")>
<WebMethod(MessageName = "AddBlog", Description = "")>
Public Function AddBlog(ParentFolderID As Integer, 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 Integer
```

Parameters

Parameters	Description
ParentFolderID	
BlogName	Name of the Blog.
BlogTitle	Title of the Blog.
BlogDescription	Description of the Blog.

Visibility	Is the blog public/private?
EnableComments	Allow commenting.
ModerateComments	If commenting is allowed, moderate comments.
RequireAuthentication	If commenting is allowed, require authentication to add comment.
Categories	String Array representing the categories for this blog.
RollLinks	The Blog's roll links.

Returns

Integer 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddCommentForPost", Description = "")]
public bool AddCommentForPost(BlogComment bcComment);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddCommentForPost", Description = "")>
Public Function AddCommentForPost(bcComment As BlogComment) As bool
```

Parameters

Parameters	Description
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 (see page 337). Return new post ID.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddPost", Description = "")]
public int AddPost(int blogID, ContentData PostContent, string[] Categories, bool Pingback,
string Tags, string TrackbackURL);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddPost", Description = "")>
Public Function AddPost(blogID As Integer, PostContent As ContentData, Categories As string[],
Pingback As bool, Tags As string, TrackbackURL As string) As Integer
```

Parameters

Parameters	Description
blogID	Integer ID of the blog to add the post to.
PostContent	ContentData class representing the HTML, title, etc.
Categories	String Array representing the categories this post applies to.
Pingback	Pingback URL(s) in this post?
Tags	String list of tags that that applies to this post.
TrackbackURL	Trackback URL(s) to notify.

Returns

Integer 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteBlog", Description = "")]
public void DeleteBlog(int BlogID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteBlog", Description = "")>
Public Function DeleteBlog(BlogID As Integer) As void
```

Parameters

Parameters	Description
BlogID	The integer ID of the blog. This corresponds to the folder id of the blog.

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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeletePost", Description = "")]
public bool DeletePost(int post_id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeletePost", Description = "")>
Public Function DeletePost(post_id As Integer) As bool
```

Parameters

Parameters	Description
post_id	Integer value of the post ID you wish to delete.

Returns

Boolean

Blog.GetBlankBlogPostData Method

Returns a blank BlogPostData class. Used Primarily in the Workarea.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBlankBlogPostData", Description = "")]
public BlogPostData GetBlankBlogPostData();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetBlankBlogPostData", Description = "")>
Public Function GetBlankBlogPostData() As BlogPostData
```

Returns

A blank, but not null, BlogPostData class.

GetBlog Method

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.

Overload List

	Name	Description
	Blog.GetBlog (int, string, int, int) (see page 565)	Returns a BlogData class for a given blog id, with parameters. This class contains the content (see page 337) for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. This function is overloaded. See Also: Blog.GetBlog Method (string, string, int, int) (see page 566)
	Blog.GetBlog (string, string, int, int) (see page 566)	Returns a BlogData class for a given path, with parameters. This class contains the content (see page 337) for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. This function is overloaded. See Also: Ektron.Services.Blog.GetBlog@int@string@int@int (see page 565)

Blog.GetBlog Method (int, string, int, int)

Returns a BlogData class for a given blog id, with parameters. This class contains the content (see page 337) for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

This function is overloaded. See Also:

Blog.GetBlog Method (string, string, int, int) (see page 566)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBlog by ID, StartDate, MaxEntries and LanguageID", Description =
"")]
public BlogData GetBlog(int ID, string StartDate, int MaxEntries, int LanguageID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetBlog by ID, StartDate, MaxEntries and LanguageID", Description =
"")>
Public Function GetBlog(ID As Integer, StartDate As string, MaxEntries As Integer, LanguageID
As Integer) As BlogData
```

Parameters

Parameters	Description
ID	The id of the blog. This is a integer value that corresponds to the folder id of the blog.
StartDate	The date to get entries from. Used if you want to get historical information. Gets information only for the date specified.
MaxEntries	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 (see page 337) is not returned, only the post info for the content (see page 337).
LanguageID	The language ID for the content (see page 337) 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.

Blog.GetBlog Method (string, string, int, int)

Returns a BlogData class for a given path, with parameters. This class contains the content (see page 337) for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

This function is overloaded. See Also:

Ektron.Services.Blog.GetBlog@int@string@int@int (see page 565)

C#

```
[SoapHeader("AuthHeader")]
[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")>
<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
`path`	The path of the blog. This is a string value that corresponds to the folder path of the blog.
`StartDate`	The date to get entries from. Used if you want to get historical information. Gets information only for the date specified.
`MaxEntries`	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 (see page 337) is not returned, only the post info for the content (see page 337).
`LanguageID`	The language ID for the content (see page 337) to get. 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

Overloaded Method to obtain a BlogData Class.

Overload List

	Name	Description
	Blog.GetBlogData (FolderData) (see page 567)	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: Ektron.Services.Blog.GetBlogData@int (see page 568) Ektron.Services.Blog.GetBlogData@int@DateTime (see page 568) Ektron.Services.Blog.GetBlogData@int@DateTime@int@int (see page 569)
	Blog.GetBlogData (int) (see page 568)	Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog. Returns either a populated BlogData class or null (nothing). This function is overloaded. See Also: Ektron.Services.Blog.GetBlogData@FolderData (see page 567) Ektron.Services.Blog.GetBlogData@int@DateTime (see page 568) Ektron.Services.Blog.GetBlogData@int@DateTime@int@int (see page 569)
	Blog.GetBlogData (int, DateTime) (see page 568)	Overloaded Method to obtain a BlogData Class. Accepts the integer 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: Ektron.Services.Blog.GetBlogData@FolderData (see page 567) Ektron.Services.Blog.GetBlogData@int (see page 568) Ektron.Services.Blog.GetBlogData@int@DateTime@int@int (see page 569)
	Blog.GetBlogData (int, DateTime, int, int) (see page 569)	Overloaded Method to obtain a BlogData Class. Accepts the integer 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: Ektron.Services.Blog.GetBlogData@FolderData (see page 567) Ektron.Services.Blog.GetBlogData@int (see page 568) Ektron.Services.Blog.GetBlogData@int@DateTime (see page 568)

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:

Ektron.Services.Blog.GetBlogData@int (see page 568)

Ektron.Services.Blog.GetBlogData@int@DateTime (see page 568)

Ektron.Services.Blog.GetBlogData@int@DateTime@int@int (see page 569)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBlogData By FolderData", Description = "")]
public BlogData GetBlogData(FolderData folder_data);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetBlogData By FolderData", Description = "")>
Public Function GetBlogData(folder_data As FolderData) As BlogData
```

Parameters

Parameters	Description
folder_data	A populated FolderData class. (see FolderData)

Returns

A populated BlogData class or nothing

Remarks

Blog.GetBlogData Method (int)

Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

Ektron.Services.Blog.GetBlogData@FolderData (see page 567)

Ektron.Services.Blog.GetBlogData@int@DateTime (see page 568)

Ektron.Services.Blog.GetBlogData@int@DateTime@int@int (see page 569)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBlogData By ID", Description = "")]
public BlogData GetBlogData(int BlogID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetBlogData By ID", Description = "")>
Public Function GetBlogData(BlogID As Integer) As BlogData
```

Parameters

Parameters	Description
BlogID	Integer 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 (int, DateTime)

Overloaded Method to obtain a BlogData Class. Accepts the integer 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:

Ektron.Services.Blog.GetBlogData@FolderData (see page 567)

Ektron.Services.Blog.GetBlogData@int (see page 568)

Ektron.Services.Blog.GetBlogData@int@DateTime@int@int (see page 569)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBlogData By ID and StartDate", Description = "")]
```

```
public BlogData GetBlogData(int BlogID, DateTime StartDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetBlogData By ID and StartDate", Description = "")>
Public Function GetBlogData(BlogID As Integer, StartDate As DateTime) As BlogData
```

Parameters

Parameters	Description
BlogID	Integer ID of the blog. This corresponds to the id of the blog folder.
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 (int, DateTime, int, int)

Overloaded Method to obtain a BlogData Class. Accepts the integer 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:

Ektron.Services.Blog.GetBlogData@FolderData (see page 567)

Ektron.Services.Blog.GetBlogData@int (see page 568)

Ektron.Services.Blog.GetBlogData@int@DateTime (see page 568)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBlogData by ID, StartDate, PostsVisible and LanguageID",
Description = "")]
public BlogData GetBlogData(int BlogID, DateTime StartDate, int Postsvisible, int LanguageID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetBlogData by ID, StartDate, PostsVisible and LanguageID",
Description = "")>
Public Function GetBlogData(BlogID As Integer, StartDate As DateTime, Postsvisible As Integer,
LanguageID As Integer) As BlogData
```

Parameters

Parameters	Description
BlogID	Integer ID of the blog. This corresponds to the id of the blog folder.
StartDate	The DateTime value to get posts from. You can obtain historical posts with this property, or leave it blank to begin now.
Postsvisible	Integer number of posts to return. Leave as 0 to use the blog setting, or -1 to return none.
LanguageID	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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBlogDataForPost", Description = "")]
public BlogData GetBlogDataForPost(int post_id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetBlogDataForPost", Description = "")>
Public Function GetBlogDataForPost(post_id As Integer) As BlogData
```

Parameters

Parameters	Description
post_id	Integer 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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBlogPostData", Description = "")]
public BlogPostData GetBlogPostData(int id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetBlogPostData", Description = "")>
Public Function GetBlogPostData(id As Integer) As BlogPostData
```

Parameters

Parameters	Description
id	Integer value of the post ID

Returns

BlogPostData Class.

GetBlogRoll Method

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 (int) (see page 571)	Returns a populated blog roll class. Accepts a blog ID integer value. See the blogroll class for more information. This function is overloaded. See Also: Ektron.Services.Blog.GetBlogRoll@string (see page 571)
	Blog.GetBlogRoll (string) (see page 571)	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 40) class for the properties and usage of the blog roll. This is an overloaded method. See Also: Ektron.Services.Blog.GetBlogRoll@int (see page 571).

Blog.GetBlogRoll Method (int)

Returns a populated blog roll class. Accepts a blog ID integer value. See the blogroll class for more information.

This function is overloaded. See Also:

Ektron.Services.Blog.GetBlogRoll@string (see page 571)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBlogRoll", Description = "")]
public BlogRoll GetBlogRoll(int blog_id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetBlogRoll", Description = "")>
Public Function GetBlogRoll(blog_id As Integer) As BlogRoll
```

Parameters

Parameters	Description
`blog_id`	Integer ID of the blog. This corresponds to the folder ID of the blog.

Returns

A populated BlogRoll (see page 40) 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 40) class for the properties and usage of the blog roll.

This is an overloaded method. See Also:

Ektron.Services.Blog.GetBlogRoll@int (see page 571).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBlogRoll By Path", Description = "")]
public BlogRoll GetBlogRoll(string Path);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetBlogRoll By Path", Description = "")>
Public Function GetBlogRoll(Path As string) As BlogRoll
```

Parameters

Parameters	Description
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 40) 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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetPost", Description = "")]
public ContentData GetPost(int post_id, BlogPostData PostInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetPost", Description = "")>
Public Function GetPost(post_id As Integer, PostInfo As BlogPostData) As ContentData
```

Parameters

Parameters	Description
post_id	Integer ID for the post.
PostInfo	BlogPostData class representing the blog specific post info. This is also in the metadata.

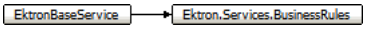
Returns

ContentData class with a ByRef BlogPostData

BusinessRules Class

Summary description for BusinessRules

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

Members

BusinessRules Methods

	Name	Description
	GetBusinessRules (see page 573)	Method to get the results from a business rule. This method is overloaded.

Topics

Name	Description
BusinessRules Methods (see page 573)	The methods of the BusinessRules class are listed here.

Legend

	Method

BusinessRules Methods

	Name	Description
	GetBusinessRules (see page 573)	Method to get the results from a business rule. This method is overloaded.

Legend

	Method

BusinessRules Methods

The methods of the BusinessRules class are listed here.

Public Methods

	Name	Description
	GetBusinessRules (see page 573)	Method to get the results from a business rule. This method is overloaded.

Legend

	Method

GetBusinessRules Method

Method to get the results from a business rule. This method is overloaded.

Overload List

	Name	Description
	BusinessRules.GetBusinessRules (string, string, string, string, string, string, string, int) (see page 573)	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.
	BusinessRules.GetBusinessRules (string, string, string, string, string, string, string, string) (see page 574)	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.

BusinessRules.GetBusinessRules Method (string, string, string, string, string, string, string, int)

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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBusinessRules By Int RuleID", Description = "")]
public string GetBusinessRules(string httpHost, string urlRest, string cookies, string
formValues, string queryValues, string serverValues, string paramValues, int RuleID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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 Integer) As string
```

Parameters

Parameters	Description
httpHost	The HTTP host the request is made on.
urlRest	not used
cookies	Cookie Value Collection (see page 63)
formValues	Form Value Collection (see page 63)
queryValues	Querystring Value Collection (see page 63)
serverValues	Server Variables Collection (see page 63)
paramValues	Custom Value Collection (see page 63)
RuleID	Integer 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.

C#

```
[SoapHeader("AuthHeader")]
[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")>
<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
httpHost	The HTTP host the request is made on.
urlRest	not used
cookies	Cookie Value Collection (see page 63)
formValues	Form Value Collection (see page 63)
queryValues	Querystring Value Collection (see page 63)
serverValues	Server Variables Collection (see page 63)
paramValues	Custom Value Collection (see page 63)

RuleID	String representing the identifier of the Ruleset. The identifier is a combination of the path/page/uniqueID of the business rules control.

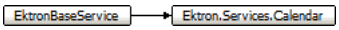
Returns

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
```

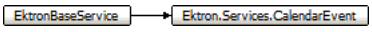
File

Calendar.cs

CalendarEvent Class

Summary description for CalendarEvent

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class CalendarEvent : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class CalendarEvent
Inherits EktronBaseService
```

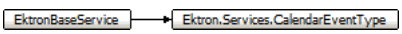
File

CalendarEvent.cs

CalendarEventType Class

Summary description for CalendarEventType

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class CalendarEventType : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class CalendarEventType
Inherits EktronBaseService
```

File

CalendarEventType.cs

Content Class

Summary description for ContentService

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)]

Members

Content Methods

	Name	Description
	AddContent (see page 579)	Adds Content to the CMS. This method is overloaded.
	CheckOutContent (see page 581)	This will check the content (see page 337) out to the current user if the content (see page 337) is currently checked in and the user has privileges to check it out.
	CopyContentToFolder (see page 581)	Copy the content (see page 337) into required folder.
	DeleteContentItem (see page 582)	Delete the content (see page 337) by the content (see page 337)'s ID.
	GetAddViewLanguage (see page 582)	Loads all the languages from the system for a given content (see page 337) ID and populates the Add/View drop down box with flag set on each item.
	GetAllComments (see page 583)	Loads all of the comments.
	GetAllTemplates (see page 583)	Loads all of the templates.
	GetAllUnassignedItemApprovals (see page 583)	Returns all of the unassigned user/groups for a given content (see page 337) ID.
	GetChildContent (see page 584)	Loads all of the contents for the given folder.
	GetContent (see page 584)	Get Content By ID
	GetContentByHistory (see page 584)	Retrieves the corresponding content (see page 337) by the history ID.

	GetContentForEditing (see page 585)	Loads the content (see page 337) details with checkout mode.
	GetContentState (see page 585)	Loads the content (see page 337)'s details.
	GetContentStatus (see page 585)	This will return a single letter string which is the status of the piece of content (see page 337). The letters returned are the same letters used in the Workarea view.
	GetDomain (see page 586)	Gets web domain for a piece of content (see page 337).
	MoveContentToFolder (see page 586)	Moves content (see page 337) to a folder.
	PublishContent (see page 586)	Function takes in information about the content (see page 337) to be checked in and published. This goes through the normal approval chain
	RenameContent (see page 587)	Renames the content (see page 337).
	SaveContent (see page 587)	Updates the checked out content (see page 337) 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 588)	Delete the content (see page 337) by the content (see page 337)'s Id and folder Id.
	UndoCheckout (see page 588)	This undoes the checkout for a piece of content (see page 337) which is checked out to the current user.
	UpdateContentMetaData (see page 588)	Updates a specific metadata item attached to a specified piece of content (see page 337)

Topics

Name	Description
Content Methods (see page 578)	The methods of the Content class are listed here.

Legend

	Method

Content Methods

	Name	Description
	AddContent (see page 579)	Adds Content to the CMS. This method is overloaded.
	CheckOutContent (see page 581)	This will check the content (see page 337) out to the current user if the content (see page 337) is currently checked in and the user has privileges to check it out.
	CopyContentToFolder (see page 581)	Copy the content (see page 337) into required folder.
	DeleteContentItem (see page 582)	Delete the content (see page 337) by the content (see page 337)'s ID.
	GetAddViewLanguage (see page 582)	Loads all the languages from the system for a given content (see page 337) ID and populates the Add/View drop down box with flag set on each item.
	GetAllComments (see page 583)	Loads all of the comments.
	GetAllTemplates (see page 583)	Loads all of the templates.
	GetAllUnassignedItemApprovals (see page 583)	Returns all of the unassigned user/groups for a given content (see page 337) ID.
	GetChildContent (see page 584)	Loads all of the contents for the given folder.
	GetContent (see page 584)	Get Content By ID
	GetContentByHistory (see page 584)	Retrieves the corresponding content (see page 337) by the history ID.
	GetContentForEditing (see page 585)	Loads the content (see page 337) details with checkout mode.
	GetContentState (see page 585)	Loads the content (see page 337)'s details.
	GetContentStatus (see page 585)	This will return a single letter string which is the status of the piece of content (see page 337). The letters returned are the same letters used in the Workarea view.
	GetDomain (see page 586)	Gets web domain for a piece of content (see page 337).
	MoveContentToFolder (see page 586)	Moves content (see page 337) to a folder.

	PublishContent (see page 586)	Function takes in information about the content (see page 337) to be checked in and published. This goes through the normal approval chain
	RenameContent (see page 587)	Renames the content (see page 337).
	SaveContent (see page 587)	Updates the checked out content (see page 337) 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 588)	Delete the content (see page 337) by the content (see page 337)'s Id and folder Id.
	UndoCheckout (see page 588)	This undoes the checkout for a piece of content (see page 337) which is checked out to the current user.
	UpdateContentMetaData (see page 588)	Updates a specific metadata item attached to a specified piece of content (see page 337)

Legend

	Method

Content Methods

The methods of the Content class are listed here.

Public Methods

	Name	Description
	AddContent (see page 579)	Adds Content to the CMS. This method is overloaded.
	CheckOutContent (see page 581)	This will check the content (see page 337) out to the current user if the content (see page 337) is currently checked in and the user has privileges to check it out.
	CopyContentToFolder (see page 581)	Copy the content (see page 337) into required folder.
	DeleteContentItem (see page 582)	Delete the content (see page 337) by the content (see page 337)'s ID.
	GetAddViewLanguage (see page 582)	Loads all the languages from the system for a given content (see page 337) ID and populates the Add/View drop down box with flag set on each item.
	GetAllComments (see page 583)	Loads all of the comments.
	GetAllTemplates (see page 583)	Loads all of the templates.
	GetAllUnassignedItemApprovals (see page 583)	Returns all of the unassigned user/groups for a given content (see page 337) ID.
	GetChildContent (see page 584)	Loads all of the contents for the given folder.
	GetContent (see page 584)	Get Content (see page 576) By ID
	GetContentByHistory (see page 584)	Retrieves the corresponding content (see page 337) by the history ID.
	GetContentForEditing (see page 585)	Loads the content (see page 337) details with checkout mode.
	GetContentState (see page 585)	Loads the content (see page 337)'s details.
	GetContentStatus (see page 585)	This will return a single letter string which is the status of the piece of content (see page 337). The letters returned are the same letters used in the Workarea view.
	GetDomain (see page 586)	Gets web domain for a piece of content (see page 337).
	MoveContentToFolder (see page 586)	Moves content (see page 337) to a folder.
	PublishContent (see page 586)	Function takes in information about the content (see page 337) to be checked in and published. This goes through the normal approval chain
	RenameContent (see page 587)	Renames the content (see page 337).
	SaveContent (see page 587)	Updates the checked out content (see page 337) 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 588)	Delete the content (see page 337) by the content (see page 337)'s Id and folder Id.

	UndoCheckout (see page 588)	This undoes the checkout for a piece of content (see page 337) which is checked out to the current user.
	UpdateContentMetaData (see page 588)	Updates a specific metadata item attached to a specified piece of content (see page 337)

Legend

	Method

AddContent Method

Adds Content to the CMS. This method is overloaded.

Overload List

	Name	Description
	Content.AddContent (string, string, string, string, string, string, int, object, object, string) (see page 579)	Adds Content (see page 576) to the CMS
	Content.AddContent (string, string, string, string, string, string, int, object, object, string, int) (see page 580)	Adds Content (see page 576) to the CMS
	Content.AddContent (string, string, string, string, string, string, int, object, object, string, int, int) (see page 580)	Adds Content (see page 576) to the CMS

Content.AddContent Method (string, string, string, string, string, string, int, object, object, string)

Adds Content (see page 576) to the CMS

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddContent", Description = "Adds a piece of content.")]
public int AddContent(string ContentTitle, string ContentComment, string ContentHtml, string
SearchText, string SummaryHtml, string ContentLanguage, int FolderId, object GoLive, object
EndDate, string MetaInfoXml);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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
Integer, GoLive As object, EndDate As object, MetaInfoXml As string) As Integer
```

Parameters

Parameters	Description
ContentTitle	Title of content (see page 337) to add
ContentComment	Content (see page 576) comments
ContentHtml	Content (see page 576) HTML to display
SearchText	Search Text
SummaryHtml	Summary HTML
ContentLanguage	Content (see page 576) Language

FolderId	Folder to place the content (see page 337) in
GoLive	Go Live Date
EndDate	Content (see page 576) End Date
MetaInfoXml	Meta Info XML

Returns

Integer - ID of newly created content (see page 337)

Content.AddContent Method (string, string, string, string, string, string, int, object, object, string, int)

Adds Content (see page 576) to the CMS

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddContent With ContentID", Description = "Adds a piece of
content.")]
public int AddContent(string ContentTitle, string ContentComment, string ContentHtml, string
SearchText, string SummaryHtml, string ContentLanguage, int FolderId, object GoLive, object
EndDate, string MetaInfoXml, int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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
Integer, GoLive As object, EndDate As object, MetaInfoXml As string, ContentID As Integer) As
Integer
```

Parameters

Parameters	Description
ContentTitle	Title of content (see page 337) to add
ContentComment	Content (see page 576) comments
ContentHtml	Content (see page 576) HTML to display
SearchText	Search Text
SummaryHtml	Summary HTML
ContentLanguage	Content (see page 576) Language
FolderId	Folder to place the content (see page 337) in
GoLive	Go Live Date
EndDate	Content (see page 576) End Date
MetaInfoXml	Meta Info XML
ContentID	Content (see page 576) ID

Returns

Integer - ID of newly created content (see page 337)

Content.AddContent Method (string, string, string, string, string, string, int, object, object, string, int, int)

Adds Content (see page 576) to the CMS

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddContent with XmlID and TemplateID", Description = "Adds a piece
of content.")]
public int AddContent(string ContentTitle, string ContentComment, string ContentHtml, string
SearchText, string SummaryHtml, string ContentLanguage, int FolderId, object GoLive, object
EndDate, string MetaInfoXml, int XmlId, int TemplateID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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
Integer, GoLive As object, EndDate As object, MetaInfoXml As string, XmlId As Integer,
TemplateID As Integer) As Integer
```

Parameters

Parameters	Description
ContentTitle	Title of content (see page 337) to add
ContentComment	Content (see page 576) comments
ContentHtml	Content (see page 576) HTML to display
SearchText	Search Text
SummaryHtml	Summary HTML
ContentLanguage	Content (see page 576) Language
FolderId	Folder to place the content (see page 337) in
GoLive	Go Live Date
EndDate	Content (see page 576) End Date
XmlId	XmlId
TemplateID	TemplateID

Returns

Integer - ID of newly created content (see page 337)

Content.CheckOutContent Method

This will check the content (see page 337) out to the current user if the content (see page 337) is currently checked in and the user has privileges to check it out.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "CheckOutContent", Description = "Checks the content out to the
current user")]
public bool CheckOutContent(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "CheckOutContent", Description = "Checks the content out to the
current user")>
Public Function CheckOutContent(ContentID As Integer) As bool
```

Parameters

Parameters	Description
ContentId	Id of the content (see page 337) to check out

Returns

A Boolean representing the success of the check out

Content.CopyContentToFolder Method

Copy the content (see page 337) into required folder.

C#

```
[SoapHeader("AuthHeader")]
[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")>
<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 576) ID
FID	Folder ID
Lang	Language ID

Returns

Ektron.Cms.ContentData

Content.DeleteContentItem Method

Delete the content (see page 337) by the content (see page 337)'s ID.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteContentItem", Description = "Delete the content by the
content's ID")]
public bool DeleteContentItem(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteContentItem", Description = "Delete the content by the
content's ID")>
Public Function DeleteContentItem(ContentID As Integer) As bool
```

Parameters

Parameters	Description
ContentID	The content (see page 337)'s ID

Content.GetAddViewLanguage Method

Loads all the languages from the system for a given content (see page 337) ID and populates the Add/View drop down box with flag set on each item.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAddViewLanguage", Description = "Gets all the languages from the
system for a given content ID ")]
public LanguageData[] GetAddViewLanguage(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAddViewLanguage", Description = "Gets all the languages from the
system for a given content ID ")>
Public Function GetAddViewLanguage(ContentID As Integer) As LanguageData[]
```

Parameters

Parameters	Description
ContentID	The Content (see page 576)'s ID

Content.GetAllComments Method

Loads all of the comments.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllComments", Description = "Gets all comments")]
public CommentData[] GetAllComments(int KeyID, int CommentID, int RefID, string RefType, int
UserID, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllComments", Description = "Gets all comments")>
Public Function GetAllComments(KeyID As Integer, CommentID As Integer, RefID As Integer,
RefType As string, UserID As Integer, OrderBy As string) As CommentData[]
```

Parameters

Parameters	Description
KeyID	key ID
CommentID	Comment ID
RefID	Reference ID, (content (see page 337) or forms)
RefType	Reference type, C = content (see page 337) or T = Task
UserID	The user ID
OrderBy	The order of the comments

Content.GetAllTemplates Method

Loads all of the templates.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllTemplates", Description = "Gets all templates")]
public TemplateData[] GetAllTemplates(string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllTemplates", Description = "Gets all templates")>
Public Function GetAllTemplates(OrderBy As string) As TemplateData[]
```

Parameters

Parameters	Description
OrderBy	Order by clause

Content.GetAllUnassignedItemApprovals Method

Returns all of the unassigned user/groups for a given content (see page 337) ID.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllUnassignedItemApprovals", Description = "Returns all of the
unassigned user/groups for a given content ID")]
public ApprovalData[] GetAllUnassignedItemApprovals(int ContentID, string ItemType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllUnassignedItemApprovals", Description = "Returns all of the
unassigned user/groups for a given content ID")>
Public Function GetAllUnassignedItemApprovals(ContentID As Integer, ItemType As string) As
ApprovalData[]
```

Parameters

Parameters	Description
ContentID	The content (see page 337)'s ID
ItemType	The type of user

Content.GetChildContent Method

Loads all of the contents for the given folder.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetChildContent", Description = "Loads all of the contents for the
given folder")]
public ContentData[] GetChildContent(int FolderID, bool Recursive, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetChildContent", Description = "Loads all of the contents for the
given folder")>
Public Function GetChildContent(FolderID As Integer, Recursive As bool, OrderBy As string) As
ContentData[]
```

Parameters

Parameters	Description
FolderID	the folders ID
Recursive	True = recursive look up. False = none
OrderBy	An order by clause

Content.GetContent Method

Get Content (see page 576) By ID

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetContent", Description = "Gets the content details by content id")]
public ContentData GetContent(int ContentID, Ektron.Cms.Content.EkContent.ContentResultType
ResultType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetContent", Description = "Gets the content details by content id")>
Public Function GetContent(ContentID As Integer, ResultType As
Ektron.Cms.Content.EkContent.ContentResultType) As ContentData
```

Parameters

Parameters	Description
ContentID	Content (see page 576) ID
Preview	Preview
Published	Is Published

Content.GetContentByHistory Method

Retrieves the corresponding content (see page 337) by the history ID.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetContentByHistory", Description = "Retrieves the corresponding
content by the history ID")]
public ContentData GetContentByHistory(int HistoryID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetContentByHistory", Description = "Retrieves the corresponding
content by the history ID")>
Public Function GetContentByHistory(HistoryID As Integer) As ContentData
```

Parameters

Parameters	Description
HistoryID	The history ID

Content.GetContentForEditing Method

Loads the content (see page 337) details with checkout mode.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetContentForEditing", Description = "Loads the content details with
checkout mode")]
public ContentEditData GetContentForEditing(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetContentForEditing", Description = "Loads the content details with
checkout mode")>
Public Function GetContentForEditing(ContentID As Integer) As ContentEditData
```

Parameters

Parameters	Description
ContentID	The content (see page 337)'s ID

Content.GetContentState Method

Loads the content (see page 337)'s details.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetContentState", Description = "Gets the content's details")]
public ContentStateData GetContentState(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetContentState", Description = "Gets the content's details")>
Public Function GetContentState(ContentID As Integer) As ContentStateData
```

Parameters

Parameters	Description
ContentID	The content (see page 337)'s ID

Content.GetContentStatus Method

This will return a single letter string which is the status of the piece of content (see page 337). The letters returned are the same letters used in the Workarea view.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetContentStatus", Description = "Gets single letter string which is
the status of the piece of content")]
public string GetContentStatus(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetContentStatus", Description = "Gets single letter string which is
the status of the piece of content")>
Public Function GetContentStatus(ContentID As Integer) As string
```

Parameters

Parameters	Description
ContentID	Id of the content (see page 337) to return the status of

Content.GetDomain Method

Gets web domain for a piece of content (see page 337).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetDomain", Description = "Gets web domain for a piece of content")]
public string GetDomain(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetDomain", Description = "Gets web domain for a piece of content")>
Public Function GetDomain(ContentID As Integer) As string
```

Parameters

Parameters	Description
ContentID	The content (see page 337)'s ID

Content.MoveContentToFolder Method

Moves content (see page 337) to a folder.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "MoveContentToFolder", Description = "Moves content to a folder")]
public ContentData MoveContentToFolder(string ContentID, string ContentLanguage, long
FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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
ContentID	The content (see page 337)'s ID
ContentLanguage	The content (see page 337)'s language
FolderID	The folder's ID

Content.PublishContent Method

Function takes in information about the content (see page 337) to be checked in and published. This goes through the normal approval chain

C#

```
[SoapHeader("AuthHeader")]
[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(int ContentID, int FolderID, int ContentLanguage, string
DontCreateTask, int UserID, string TaskTitle);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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 Integer, FolderID As Integer, ContentLanguage As
Integer, DontCreateTask As string, UserID As Integer, TaskTitle As string) As bool
```

Parameters

Parameters	Description
ContentID	Integer value of the content (see page 337)'s ID
FolderID	Integer value of the content (see page 337) folder's ID
ContentLanguage	Integer value of the content (see page 337)'s language ID
DontCreateTask	If this parameter is "" a task will be created for this submit, else it will not
UserID	The user ID of task assigned by (current user). This is needed only if "dontCreateTask" is "", it is ignored else
TaskTitle	Title for the task. This is needed only if "dontCreateTask" is "", it is ignored else

Content.RenameContent Method

Renames the content (see page 337).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "RenameContent", Description = "Renames the content by ID")]
public ContentData RenameContent(int ContentID, string Title);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "RenameContent", Description = "Renames the content by ID")>
Public Function RenameContent(ContentID As Integer, Title As string) As ContentData
```

Parameters

Parameters	Description
ContentID	The content (see page 337)'s ID
Title	The content (see page 337)'s new title

Content.SaveContent Method

Updates the checked out content (see page 337) block. Only following properties can be updated: contentData.Title contentData.Teaser contentData.Comment contentData.GoLive contentData.Html contentData.EndDate contentData.EndDateAction contentData.ManualAlias

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "SaveContent", Description = "Updates the checked out content
block.")]
public bool SaveContent(ContentEditData contentEditData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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 576) must be in the checkedout state.

Content.SubmitForDelete Method

Delete the content (see page 337) by the content (see page 337)'s Id and folder Id.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "SubmitForDelete", Description = "Deletes the content by the
content's Id and folder Id")]
public bool SubmitForDelete(int ContentID, int FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "SubmitForDelete", Description = "Deletes the content by the
content's Id and folder Id")>
Public Function SubmitForDelete(ContentID As Integer, FolderID As Integer) As bool
```

Parameters

Parameters	Description
ContentID	The content (see page 337)'s Id
FolderID	The content (see page 337)'s Folder Id

Content.UndoCheckout Method

This undoes the checkout for a piece of content (see page 337) which is checked out to the current user.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UndoCheckout", Description = "Reverts the checkout for a piece of
content which is checked out to the current user")]
public bool UndoCheckout(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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 Integer) As bool
```

Parameters

Parameters	Description
ContentID	Id of the content (see page 337) for which to undo check out

Content.UpdateContentMetaData Method

Updates a specific metadata item attached to a specified piece of content (see page 337)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateContentMetaData", Description = "Updates Content Metadata")]
```

```
public bool UpdateContentMetaData(int ContentID, int MetaDataTypeID, string MetaDataText);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UpdateContentMetaData", Description = "Updates Content Metadata")>
Public Function UpdateContentMetaData(ContentID As Integer, MetaDataTypeID As Integer,
MetaDataText As string) As bool
```

ContentRating Class

Summary description for ContentRating

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

Members

ContentRating Methods

	Name	Description
	AddContentRating (see page 590)	Adds a content (see page 337) rating to the database. Internally a check is done to ensure the user has not already rated the content (see page 337), and passes down the rating and any comments the user supplied.
	GetContentRatingResults (see page 590)	Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content (see page 337) from the range of StartDate to EndDate. In order to get the full range of ratings, pass DateTime.Min and DateTime.Max as arguments.
	PurgeContentRatings (see page 591)	Purges all ratings of a particular content (see page 337) block over a defined date/time range.

Topics

Name	Description
ContentRating Methods (see page 590)	The methods of the ContentRating class are listed here.

Legend

	Method

ContentRating Methods

	Name	Description
	AddContentRating (see page 590)	Adds a content (see page 337) rating to the database. Internally a check is done to ensure the user has not already rated the content (see page 337), and passes down the rating and any comments the user supplied.
	GetContentRatingResults (see page 590)	Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content (see page 337) from the range of StartDate to EndDate. In order to get the full range of ratings, pass DateTime.Min and DateTime.Max as arguments.
	PurgeContentRatings (see page 591)	Purges all ratings of a particular content (see page 337) block over a defined date/time range.

Legend

	Method

ContentRating Methods

The methods of the ContentRating class are listed here.

Public Methods

	Name	Description
	AddContentRating (see page 590)	Adds a content (see page 337) rating to the database. Internally a check is done to ensure the user has not already rated the content (see page 337), and passes down the rating and any comments the user supplied.
	GetContentRatingResults (see page 590)	Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content (see page 337) from the range of StartDate to EndDate. In order to get the full range of ratings, pass DateTime.Min and DateTime.Max as arguments.
	PurgeContentRatings (see page 591)	Purges all ratings of a particular content (see page 337) block over a defined date/time range.

Legend

	Method

ContentRating.AddContentRating Method

Adds a content (see page 337) rating to the database. Internally a check is done to ensure the user has not already rated the content (see page 337), and passes down the rating and any comments the user supplied.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddContentRating", Description = "")]
public void AddContentRating(int contentid, int userid, int userRating, string userComments);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddContentRating", Description = "")>
Public Function AddContentRating(contentid As Integer, userid As Integer, userRating As
Integer, userComments As string) As void
```

Parameters

Parameters	Description
contentid	The numeric ID associated with the content (see page 337) block being rated.
userid	The unique global identifier stored as a cookie by the user.
userRating	The rating the user supplies (1-9)
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 (see page 337) 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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetContentRatingResults", Description = "")]
public System.Data.DataSet GetContentRatingResults(int contentID, DateTime startDate, DateTime
endDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetContentRatingResults", Description = "")>
Public Function GetContentRatingResults(contentID As Integer, startDate As DateTime, endDate
As DateTime) As System.Data.DataSet
```

Parameters

Parameters	Description
contentID	The numeric ID associated with the content (see page 337) block whose ratings are being queried.
startDate	The start date of a range of dates to query over.
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 (see page 337) block over a defined date/time range.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "PurgeContentRatings", Description = "")]
public void PurgeContentRatings(int contentid, DateTime startDate, DateTime endDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "PurgeContentRatings", Description = "")>
Public Function PurgeContentRatings(contentid As Integer, startDate As DateTime, endDate As
DateTime) As void
```

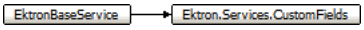
Parameters

Parameters	Description
contentid	The ID of the content (see page 337) that is having its ratings purged.
startDate	The beginning of the date range of content (see page 337) to be purged.
endDate	The end of the date range of content (see page 337) to be purged.

CustomFields Class

Summary description for CustomFields

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

Members

CustomFields Methods

	Name	Description
	AddNewEditableCustomFieldAssignments (see page 594)	Returns the HTML string for the add/edit custom fields.

	BoolToYesNo (see page 595)	Returns labels for yes or no based on locale.
	GetEditableCustomFieldAssignments (see page 595)	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 -> going unchecked, then ensure required us unchecked. • When clicked: if 'Required' button, don't accept checks if assigned is unchecked. • When form submitted, copy checked... more (see page 595)
	GetEnhancedMetadataArea (see page 596)	Returns the HTML needed to render the EnhancedMetadata selection area.
	GetEnhancedMetadataScript (see page 596)	Returns support javascript code for enhanced metadata.
	GetFieldsByFolder (see page 597)	Returns all of the custom attributes by folder.
	GetSearchProperties (see page 597)	Returns an HTML string for the search properties display.
	ProcessCustomFields (see page 597)	Process Folders' Custom-Field (and metadata) assignments: When form submitted, copy checked items to hidden field. Form 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.
	SearchAllAssets (see page 598)	Returns the search result based on assets.
	WriteFilteredMetadataForEdit (see page 598)	Returns an HTML string for filtered metadata.
	WriteFilteredMetadataForView (see page 599)	Returns an HTML string for metadata view.
	WriteLibrarySearchExtended (see page 599)	Returns an HTML string for the library search custom metadata properties.
	WriteMetadataDefaultForEdit (see page 599)	Returns an HTML string for the default metadata.
	WriteMetadataTypeForView (see page 600)	Returns a metadata type for view.
	WritePublicSearch (see page 600)	Returns WritepublicsearchEx.

Topics

Name	Description
CustomFields Methods (see page 593)	The methods of the CustomFields class are listed here.

Legend

	Method

CustomFields Methods

	Name	Description
	AddNewEditableCustomFieldAssignments (see page 594)	Returns the HTML string for the add/edit custom fields.
	BoolToYesNo (see page 595)	Returns labels for yes or no based on locale.

	GetEditableCustomFieldAssignments (see page 595)	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 -> going unchecked, then ensure required us unchecked. • When clicked: if 'Required' button, don't accept checks if assigned is unchecked. • When form submitted, copy checked... more (see page 595)
	GetEnhancedMetadataArea (see page 596)	Returns the HTML needed to render the EnhancedMetadata selection area.
	GetEnhancedMetadataScript (see page 596)	Returns support javascript code for enhanced metadata.
	GetFieldsByFolder (see page 597)	Returns all of the custom attributes by folder.
	GetSearchProperties (see page 597)	Returns an HTML string for the search properties display.
	ProcessCustomFields (see page 597)	Process Folders' Custom-Field (and metadata) assignments: When form submitted, copy checked items to hidden field. Form 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.
	SearchAllAssets (see page 598)	Returns the search result based on assets.
	WriteFilteredMetadataForEdit (see page 598)	Returns an HTML string for filtered metadata.
	WriteFilteredMetadataForView (see page 599)	Returns an HTML string for metadata view.
	WriteLibrarySearchExtended (see page 599)	Returns an HTML string for the library search custom metadata properties.
	WriteMetadataDefaultForEdit (see page 599)	Returns an HTML string for the default metadata.
	WriteMetadataTypeForView (see page 600)	Returns a metadata type for view.
	WritePublicSearch (see page 600)	Returns WritepublicsearchEx.

Legend

	Method

CustomFields Methods

The methods of the CustomFields class are listed here.

Public Methods

	Name	Description
	AddNewEditableCustomFieldAssignments (see page 594)	Returns the HTML string for the add/edit custom fields.
	BoolToYesNo (see page 595)	Returns labels for yes or no based on locale.

	GetEditableCustomFieldAssignments (see page 595)	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 -> going unchecked, then ensure required us unchecked. • When clicked: if 'Required' button, don't accept checks if assigned is unchecked. • When form submitted, copy checked... more (see page 595)
	GetEnhancedMetadataArea (see page 596)	Returns the HTML needed to render the EnhancedMetadata selection area.
	GetEnhancedMetadataScript (see page 596)	Returns support javascript code for enhanced metadata.
	GetFieldsByFolder (see page 597)	Returns all of the custom attributes by folder.
	GetSearchProperties (see page 597)	Returns an HTML string for the search properties display.
	ProcessCustomFields (see page 597)	Process Folders' Custom-Field (and metadata) assignments: When form submitted, copy checked items to hidden field. Form 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.
	SearchAllAssets (see page 598)	Returns the search result based on assets.
	WriteFilteredMetadataForEdit (see page 598)	Returns an HTML string for filtered metadata.
	WriteFilteredMetadataForView (see page 599)	Returns an HTML string for metadata view.
	WriteLibrarySearchExtended (see page 599)	Returns an HTML string for the library search custom metadata properties.
	WriteMetadataDefaultForEdit (see page 599)	Returns an HTML string for the default metadata.
	WriteMetadataTypeForView (see page 600)	Returns a metadata type for view.
	WritePublicSearch (see page 600)	Returns WritepublicsearchEx.

Legend

	Method

CustomFields.AddNewEditableCustomFieldAssignments Method

Returns the HTML string for the add/edit custom fields.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddNewEditableCustomFieldAssignments", Description = "")]
public string AddNewEditableCustomFieldAssignments(int parentfolderid);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
```

```
<WebMethod(MessageName = "AddNewEditableCustomFieldAssignments", Description = "")>
Public Function AddNewEditableCustomFieldAssignments(parentfolderid As Integer) As string
```

Parameters

Parameters	Description
parentfolderid	The parent folder's ID.

Returns

String

CustomFields.BoolToYesNo Method

Returns labels for yes or no based on locale.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "BoolToYesNo", Description = "")]
public string BoolToYesNo(bool Value);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "BoolToYesNo", Description = "")>
Public Function BoolToYesNo(Value As bool) As string
```

Parameters

Parameters	Description
Value	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 -> going 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 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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetEditableCustomFieldAssignments", Description = "")]
public string GetEditableCustomFieldAssignments(object folderID, object editMode);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetEditableCustomFieldAssignments", Description = "")>
Public Function GetEditableCustomFieldAssignments(folderID As object, editMode As object) As
string
```

Parameters

Parameters	Description
folderID	The folder's ID.
editMode	Edit mode.

Returns

String

Remarks

Function GetEditableCustomFieldAssignments: ''

CustomFields.GetEnhancedMetadataArea Method

Returns the HTML needed to render the EnhancedMetadata selection area.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetEnhancedMetadataArea", Description = "")]
public string GetEnhancedMetadataArea();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetEnhancedMetadataArea", Description = "")>
Public Function GetEnhancedMetadataArea() As string
```

CustomFields.GetEnhancedMetadataScript Method

Returns support javascript code for enhanced metadata.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetEnhancedMetadataScript", Description = "")]
public string GetEnhancedMetadataScript();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetEnhancedMetadataScript", Description = "")>
Public Function GetEnhancedMetadataScript() As string
```

Returns

String

CustomFields.GetFieldsByFolder Method

Returns all of the custom attributes by folder.

C#

```
public Collection GetFieldsByFolder(long FolderID, long LanguageID);
```

Visual Basic

```
Public Function GetFieldsByFolder(FolderID As long, LanguageID As long) As Collection
```

Parameters

Parameters	Description
FolderID	The folder's ID.
LanguageID	The language's ID.

Returns

Microsoft.VisualBasic.Collection (see page 63)

CustomFields.GetSearchProperties Method

Returns an HTML string for the search properties display.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetSearchProperties", Description = "")]
public StringBuilder GetSearchProperties(SearchAssetDispayRequest sadReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetSearchProperties", Description = "")>
Public Function GetSearchProperties(sadReq As SearchAssetDispayRequest) As StringBuilder
```

Parameters

Parameters	Description
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 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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "ProcessCustomFields", Description = "")]
public object ProcessCustomFields(long folderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "ProcessCustomFields", Description = "")>
Public Function ProcessCustomFields(folderID As long) As object
```

Parameters

Parameters	Description
folderID	The folder's ID.

Returns

Object

CustomFields.SearchAllAssets Method

Returns the search result based on assets.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "SearchAllAssets", Description = "")]
public StringBuilder SearchAllAssets(SearchAssetDispayRequest sadReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "SearchAllAssets", Description = "")>
Public Function SearchAllAssets(sadReq As SearchAssetDispayRequest) As StringBuilder
```

Parameters

Parameters	Description
sadReq	Search parameters. (Required)

Returns

System.Text.StringBuilder

CustomFields.WriteFilteredMetadataForEdit Method

Returns an HTML string for filtered metadata.

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
MetadataTypes	Metadata (see page 626) types.
SearchableOnly	bSearchableOnly
ty	type
FolderId	The folder's ID.
Count	Number of items. (Required)

Returns

System.Text.StringBuilder

Description

[WebMethod(MessageName = "WriteFilteredMetadataForEdit", Description = "")]

CustomFields.WriteFilteredMetadataForView Method

Returns an HTML string for metadata view.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "WriteFilteredMetadataForView", Description = "")]
public string WriteFilteredMetadataForView(ContentMetaData[] MetadataTypes, long FolderId,
bool SearchableOnly);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "WriteFilteredMetadataForView", Description = "")>
Public Function WriteFilteredMetadataForView(MetadataTypes As ContentMetaData[], FolderId As
long, SearchableOnly As bool) As string
```

Parameters

Parameters	Description
MetadataTypes	Metadata (see page 626) types.
FolderId	The folder's Id.
SearchableOnly	True or False

Returns

String

CustomFields.WriteLibrarySearchExtended Method

Returns an HTML string for the library search custom metadata properties.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "WriteLibrarySearchExtended", Description = "")]
public StringBuilder WriteLibrarySearchExtended(int folderId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "WriteLibrarySearchExtended", Description = "")>
Public Function WriteLibrarySearchExtended(folderId As Integer) As StringBuilder
```

Parameters

Parameters	Description
folderId	The folder's ID.

Returns

System.Text.StringBuilder

CustomFields.WriteMetadataDefaultForEdit Method

Returns an HTML string for the default metadata.

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
DataType	Data types
cMetadataType	Metadata (see page 626) values.

Returns

System.Text.StringBuilder

Description

[WebMethod(MessageName = "WriteMetadataDefaultForEdit", Description = "")]

CustomFields.WriteMetadataTypeForView Method

Returns a metadata type for view.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "WriteMetadataTypeForView", Description = "")]
public string WriteMetadataTypeForView(string DataType, string Caption, string Value);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "WriteMetadataTypeForView", Description = "")>
Public Function WriteMetadataTypeForView(DataType As string, Caption As string, Value As
string) As string
```

Parameters

Parameters	Description
DataType	Data types.
Caption	Caption value.
Value	Value. (Required)

Returns

String

CustomFields.WritePublicSearch Method

Returns WritepublicsearchEx.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "WritePublicSearch", Description = "")]
public StringBuilder WritePublicSearch(SearchAssetDispayRequest sadReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "WritePublicSearch", Description = "")>
Public Function WritePublicSearch(sadReq As SearchAssetDispayRequest) As StringBuilder
```

Parameters

Parameters	Description
sadReq	Search asset data. (Required)

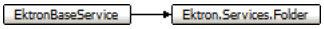
Returns

System.Text.StringBuilder

Folder Class

Summary description for Folder

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

Members

Folder Methods

	Name	Description
	AddFolder (see page 602)	Adds a folder.
	CreateFolder (see page 602)	Creates a folder.
	DeleteFolder (see page 603)	Deletes folder by folder id
	GetChildFolders (see page 603)	Loads all the subfolders for a given folder ID.
	GetFolder (see page 603)	Retrieves the folder's details.
	GetFolderID (see page 604)	
	GetPath (see page 605)	Find's a parent ID by using a given folder ID.
	RenameFolder (see page 605)	Renames folder to specified title
	UpdateFolder (see page 605)	Updates a folder.

Topics

Name	Description
Folder Methods (see page 602)	The methods of the Folder class are listed here.

Legend

	Method

Folder Methods

	Name	Description
	AddFolder (see page 602)	Adds a folder.
	CreateFolder (see page 602)	Creates a folder.
	DeleteFolder (see page 603)	Deletes folder by folder id
	GetChildFolders (see page 603)	Loads all the subfolders for a given folder ID.
	GetFolder (see page 603)	Retrieves the folder's details.
	GetFolderID (see page 604)	
	GetPath (see page 605)	Find's a parent ID by using a given folder ID.
	RenameFolder (see page 605)	Renames folder to specified title
	UpdateFolder (see page 605)	Updates a folder.

Legend

	Method

Folder Methods

The methods of the Folder class are listed here.

Public Methods

	Name	Description
	AddFolder (see page 602)	Adds a folder.
	CreateFolder (see page 602)	Creates a folder.
	DeleteFolder (see page 603)	Deletes folder by folder id
	GetChildFolders (see page 603)	Loads all the subfolders for a given folder ID.
	GetFolder (see page 603)	Retrieves the folder's details.
	GetFolderID (see page 604)	
	GetPath (see page 605)	Find's a parent ID by using a given folder ID.
	RenameFolder (see page 605)	Renames folder to specified title
	UpdateFolder (see page 605)	Updates a folder.

Legend

	Method

Folder.AddFolder Method

Adds a folder.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddFolder", Description = "Adds a folder")]
public void AddFolder(ref FolderRequest FolderReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddFolder", Description = "Adds a folder")>
Public Function AddFolder(FolderReq As ref FolderRequest) As void
```

Parameters

Parameters	Description
FolderReq	Folder Request Object

Folder.CreateFolder Method

Creates a folder.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "CreateFolder", Description = "Creates a folder with the given path")]
public bool CreateFolder(string FolderPath);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "CreateFolder", Description = "Creates a folder with the given path")>
Public Function CreateFolder(FolderPath As string) As bool
```

Parameters

Parameters	Description
FolderPath	The path on the IIS server. (Required)

Folder.DeleteFolder Method

Deletes folder by folder id

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteFolder", Description = "Deletes folder specified by folder
id")]
public bool DeleteFolder(int FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteFolder", Description = "Deletes folder specified by folder
id")>
Public Function DeleteFolder(FolderID As Integer) As bool
```

Parameters

Parameters	Description
FolderID	Folder ID

Folder.GetChildFolders Method

Loads all the subfolders for a given folder ID.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetChildFolders", Description = "Get all the subfolders for a given
folder ID")]
public FolderData[] GetChildFolders(int FolderID, bool Recursive, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetChildFolders", Description = "Get all the subfolders for a given
folder ID")>
Public Function GetChildFolders(FolderID As Integer, Recursive As bool, OrderBy As string) As
FolderData[]
```

Parameters

Parameters	Description
FolderID	The folder ID
Recursive	True = recursive look up. False = none
OrderBy	An order by clause

Folder.GetFolder Method

Retrieves the folder's details.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetFolder", Description = "Retrieves the folder's details")]
public FolderData GetFolder(int FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetFolder", Description = "Retrieves the folder's details")>
```

```
Public Function GetFolder(FolderID As Integer) As FolderData
```

Parameters

Parameters	Description
FolderID	The folder's ID

GetFolderID Method

Overload List

	Name	Description
	Folder.GetFolderID (int) (see page 604)	Finds the folder ID by using the a content (see page 337)'s ID.
	Folder.GetFolderID (string, int) (see page 604)	Finds the folder ID by folder name and in a parent folder specified.

Folder.GetFolderID Method (int)

Finds the folder ID by using the a content (see page 337)'s ID.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetFolderID", Description = "Finds the folder ID by using the a
content's ID")]
public int GetFolderID(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetFolderID", Description = "Finds the folder ID by using the a
content's ID")>
Public Function GetFolderID(ContentID As Integer) As Integer
```

Parameters

Parameters	Description
ContentID	The content (see page 337)'s ID

Folder.GetFolderID Method (string, int)

Finds the folder ID by folder name and in a parent folder specified.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetFolderIDByName", Description = "Finds the folder ID by folder
name and in a parent folder specified.")]
public int GetFolderID(string FolderName, int ParentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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 Integer) As Integer
```

Parameters

Parameters	Description
FolderName	Name of the folder
ParentId	Look in a folder

Returns

Returns the folder id. Returns 0 if it does not exist.

Folder.GetPath Method

Find's a parent ID by using a given folder ID.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetPath", Description = "Gets path to folder by folder id")]
public string GetPath(int FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetPath", Description = "Gets path to folder by folder id")>
Public Function GetPath(FolderID As Integer) As string
```

Parameters

Parameters	Description
FolderID	The folder's ID

Folder.RenameFolder Method

Renames folder to specified title

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "RenameFolder", Description = "Renames the folder by id")]
public bool RenameFolder(int FolderID, string Title);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "RenameFolder", Description = "Renames the folder by id")>
Public Function RenameFolder(FolderID As Integer, Title As string) As bool
```

Parameters

Parameters	Description
FolderID	The Folder's ID
Title	New Title

Folder.UpdateFolder Method

Updates a folder.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateFolder", Description = "Updates a folder")]
public void UpdateFolder(FolderRequest FolderReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UpdateFolder", Description = "Updates a folder")>
Public Function UpdateFolder(FolderReq As FolderRequest) As void
```

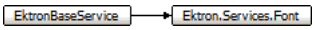
Parameters

Parameters	Description
FolderReq	Folder Request Object

Font Class

Summary description for Font

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

Members

Font Methods

	Name	Description
	AddFont (see page 607)	Adds a new font to the system.
	DeleteFont (see page 607)	Deletes a font from the system.
	GetAllFonts (see page 607)	Loads all of the fonts as a array of FontData.
	GetFont (see page 608)	Loads the required font using the font's ID.
	GetFontConfigList (see page 608)	Returns all of the font details in a drop down control.
	UpdateFont (see page 608)	Updates the font.

Topics

Name	Description
Font Methods (see page 606)	The methods of the Font class are listed here.

Legend

	Method

Font Methods

	Name	Description
	AddFont (see page 607)	Adds a new font to the system.
	DeleteFont (see page 607)	Deletes a font from the system.
	GetAllFonts (see page 607)	Loads all of the fonts as a array of FontData.
	GetFont (see page 608)	Loads the required font using the font's ID.
	GetFontConfigList (see page 608)	Returns all of the font details in a drop down control.
	UpdateFont (see page 608)	Updates the font.

Legend

	Method

Font Methods

The methods of the Font class are listed here.

Public Methods

	Name	Description
	AddFont (see page 607)	Adds a new font to the system.
	DeleteFont (see page 607)	Deletes a font from the system.
	GetAllFonts (see page 607)	Loads all of the fonts as a array of FontData.
	GetFont (see page 608)	Loads the required font using the font's ID.
	GetFontConfigList (see page 608)	Returns all of the font details in a drop down control.
	UpdateFont (see page 608)	Updates the font.

Legend

	Method

Font.AddFont Method

Adds a new font to the system.

C#

```
public void AddFont(Collection data);
```

Visual Basic

```
Public Function AddFont(data As Collection) As void
```

Parameters

Parameters	Description
data	Font data as a value key pair

Font.DeleteFont Method

Deletes a font from the system.

C#

```
public void DeleteFont(Collection data);
```

Visual Basic

```
Public Function DeleteFont(data As Collection) As void
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllFonts", Description = "")]
public FontData[] GetAllFonts();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllFonts", Description = "")>
Public Function GetAllFonts() As FontData[]
```

Returns

Ektron.Cms.FontData()

Remarks

User login is required to perform this operation.

Font.GetFont Method

Loads the required font using the font's ID.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetFont", Description = "")]
public FontData GetFont(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetFont", Description = "")>
Public Function GetFont(Id As Integer) As FontData
```

Parameters

Parameters	Description
Id	The font's ID

Returns

Integer

Font.GetFontConfigList Method

Returns all of the font details in a drop down control.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetFontConfigList", Description = "")]
public string GetFontConfigList();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetFontConfigList", Description = "")>
Public Function GetFontConfigList() As string
```

Returns

String

Font.UpdateFont Method

Updates the font.

C#

```
public void UpdateFont(Collection data);
```

Visual Basic

```
Public Function UpdateFont(data As Collection) As void
```

Parameters

Parameters	Description
data	The font's data

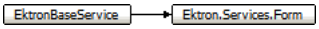
Remarks

A user with administrator privileges must be logged in to perform this operation.

Form Class

Summary description for Form

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

Members

Form Methods

	Name	Description
	AssignFormTask (see page 610)	Assigns a form submission task, as specified in FormInfo (see page 320), of the SubmittedDataXml.
	EmailFormFieldData (see page 611)	Sends an email, as specified in FormInfo (see page 320), of the SubmittedDataXml.
	GetAllForms (see page 611)	Retrieves an array of all the forms contained with the specified folder. This method is overloaded.
	GetForm (see page 612)	Retrieves information about the specified form.
	GetFormDataHistogram (see page 613)	
	GetFormFieldData (see page 613)	Retrieves the submitted data for all fields from a specified form. This method is overloaded.
	GetFormFieldList (see page 614)	Retrieves the list of form fields defined in the specified form.
	GetFormFieldQuestions (see page 615)	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 615)	Same as GetFormById, but returns only the form's title.
	SubmitFormFieldData (see page 616)	Adds form data as if it was submitted during a browser post to the web server.
	UpdateFormFieldData (see page 616)	Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

Topics

Name	Description
Form Methods (see page 610)	The methods of the Form class are listed here.

Legend

	Method

Form Methods

	Name	Description
	AssignFormTask (see page 610)	Assigns a form submission task, as specified in FormInfo (see page 320), of the SubmittedDataXml.
	EmailFormFieldData (see page 611)	Sends an email, as specified in FormInfo (see page 320), of the SubmittedDataXml.
	GetAllForms (see page 611)	Retrieves an array of all the forms contained with the specified folder. This method is overloaded.
	GetForm (see page 612)	Retrieves information about the specified form.
	GetFormDataHistogram (see page 613)	
	GetFormFieldData (see page 613)	Retrieves the submitted data for all fields from a specified form. This method is overloaded.
	GetFormFieldList (see page 614)	Retrieves the list of form fields defined in the specified form.
	GetFormFieldQuestions (see page 615)	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 615)	Same as GetFormById, but returns only the form's title.
	SubmitFormFieldData (see page 616)	Adds form data as if it was submitted during a browser post to the web server.
	UpdateFormFieldData (see page 616)	Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

Legend

	Method

Form Methods

The methods of the Form class are listed here.

Public Methods

	Name	Description
	AssignFormTask (see page 610)	Assigns a form submission task, as specified in FormInfo (see page 320), of the SubmittedDataXml.
	EmailFormFieldData (see page 611)	Sends an email, as specified in FormInfo (see page 320), of the SubmittedDataXml.
	GetAllForms (see page 611)	Retrieves an array of all the forms contained with the specified folder. This method is overloaded.
	GetForm (see page 612)	Retrieves information about the specified form.
	GetFormDataHistogram (see page 613)	
	GetFormFieldData (see page 613)	Retrieves the submitted data for all fields from a specified form. This method is overloaded.
	GetFormFieldList (see page 614)	Retrieves the list of form fields defined in the specified form.
	GetFormFieldQuestions (see page 615)	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 615)	Same as GetFormById, but returns only the form's title.
	SubmitFormFieldData (see page 616)	Adds form data as if it was submitted during a browser post to the web server.
	UpdateFormFieldData (see page 616)	Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

Legend

	Method

Form.AssignFormTask Method

Assigns a form submission task, as specified in FormInfo (see page 320), of the SubmittedDataXml.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AssignFormTask", Description = "")]
public bool AssignFormTask(FormData FormInfo, string SubmittedDataXml);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AssignFormTask", Description = "")>
Public Function AssignFormTask(FormInfo As FormData, SubmittedDataXml As string) As bool
```

Parameters

Parameters	Description
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)
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 616) 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 616).

Form.EmailFormFieldData Method

Sends an email, as specified in FormInfo (see page 320), of the SubmittedDataXml.

C#

```
public void EmailFormFieldData(FormData FormInfo, string SubmittedDataXml);
```

Visual Basic

```
Public Function EmailFormFieldData(FormInfo As FormData, SubmittedDataXml As string) As void
```

Parameters

Parameters	Description
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)
SubmittedDataXml	The form data submitted in XML format. Value Type: String (System.String)

Remarks

This routine is called automatically when calling SubmitFormFieldData (see page 616) 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 616).

GetAllForms Method

Retrieves an array of all the forms contained with the specified folder. This method is overloaded.

Overload List

	Name	Description
	Form.GetAllForms () (see page 612)	Returns all of the forms as an array of FormData.

	Form.GetAllForms (int, string, int, ContentAPI.FormResultType) (see page 612)	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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllForms", Description = "")]
public FormData[] GetAllForms();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllForms", Description = "")>
Public Function GetAllForms() As FormData[]
```

Returns

Ektron.Cms.FormData()

Remarks

User login required to perform this operation.

Form.GetAllForms Method (int, 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllForms By Id, OrderBy, LanguageID and Type", Description = "")]
public FormData[] GetAllForms(int Id, string OrderBy, int LanguageId,
ContentAPI.FormResultType Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllForms By Id, OrderBy, LanguageID and Type", Description = "")>
Public Function GetAllForms(Id As Integer, OrderBy As string, LanguageId As Integer, Type As
ContentAPI.FormResultType) As FormData[]
```

Parameters

Parameters	Description
Id	Specifies the folder.
OrderBy	Specifies how to sort the forms
LanguageId	Specifies a language or EkConstants.g_AllContentLanguages (-1) for all languages.
Type	Specifies whether only active forms or archived forms should be returned.

Remarks

Requires authentication.

Form.GetForm Method

Retrieves information about the specified form.

C#

```
[SoapHeader("AuthHeader")]
```

```
[WebMethod(MessageName = "GetForm", Description = "")]
public FormData GetForm(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetForm", Description = "")>
Public Function GetForm(Id As Integer) As FormData
```

Parameters

Parameters	Description
Id	Specifies the form.

Returns

Returns information about the form.

Remarks

Does not require authentication.

Form.GetFormDataHistogram Method

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetFormDataHistogram", Description = "")]
public ArrayList GetFormDataHistogram(long FormId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetFormDataHistogram", Description = "")>
Public Function GetFormDataHistogram(FormId As long) As ArrayList
```

Parameters

Parameters	Description
FormId	

GetFormFieldData Method

Retrieves the submitted data for all fields from a specified form. This method is overloaded.

Overload List

	Name	Description
	Form.GetFormFieldData (int, int) (see page 613)	Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.
	Form.GetFormFieldData (int, string, string) (see page 614)	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 (int, int)

Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetFormFieldData By FormID and FormDataId", Description = "")]
public FormSubmittedData GetFormFieldData(int FormId, int FormDataId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetFormFieldData By FormID and FormDataId", Description = "")>
Public Function GetFormFieldData(FormId As Integer, FormDataId As Integer) As FormSubmittedData
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)
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 (int, 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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetFormFieldData By FormId, StartDate and EndDate", Description =
"")]
public FormSubmittedData[] GetFormFieldData(int FormId, string StartDate, string EndDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetFormFieldData By FormId, StartDate and EndDate", Description =
"")>
Public Function GetFormFieldData(FormId As Integer, StartDate As string, EndDate As string) As
FormSubmittedData[]
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)
StartDate	Optionally limit the data returned to include only those submitted on or after the given date. Value Type: String (System.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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetFormFieldList", Description = "")]
public FormFieldList GetFormFieldList(int FormId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetFormFieldList", Description = "")>
Public Function GetFormFieldList(FormId As Integer) As FormFieldList
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)

Returns

The list for defined fields in the given form. Cms.FormFieldList (Ektron.Cms.FormFieldList)

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(int FormId);
```

Visual Basic

```
Public Function GetFormFieldQuestions(FormId As Integer) As Hashtable
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetFormTitle", Description = "")]
public string GetFormTitle(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetFormTitle", Description = "")>
Public Function GetFormTitle(Id As Integer) As string
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "SubmitFormFieldData", Description = "")]
public void SubmitFormFieldData(int FormId, FormSubmittedData Data, bool SendNotification,
string SubmittedDataXml);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "SubmitFormFieldData", Description = "")>
Public Function SubmitFormFieldData(FormId As Integer, Data As FormSubmittedData,
SendNotification As bool, SubmittedDataXml As string) As void
```

Parameters

Parameters	Description
FormId	Specifies the form that corresponds to the data. Value Type: Int32 (System.Int32)
Data	The data to submit. Value Type: Cms.FormSubmittedData (Ektron.Cms.FormSubmittedData)
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)
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateFormFieldData", Description = "")]
public void UpdateFormFieldData(int FormId, int FormDataId, FormSubmittedData SubmittedData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UpdateFormFieldData", Description = "")>
Public Function UpdateFormFieldData(FormId As Integer, FormDataId As Integer, SubmittedData As
FormSubmittedData) As void
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)
FormDataId	Specifies the instance of submitted data. Value Type: Int32 (System.Int32)
SubmittedData	The field data values to be updated. Value Type: Cms.FormSubmittedData (Ektron.Cms.FormSubmittedData)

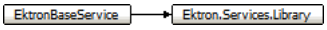
Remarks

Does not require authentication.

Library Class

Summary description for Library

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

Members

Library Methods

	Name	Description
	AddLoadBalanceItem (see page 619)	Adds a load balance path for files and images.
	DeleteLibraryItem (see page 619)	Deletes the library item by extension.
	DeleteLibraryItemById (see page 619)	Deletes the library items from the system.
	DeleteLoadBalanceItem (see page 620)	Deletes the load balance item from the system.
	GetAllChildLibItems (see page 620)	Loads all of the library items by type.
	GetAllLoadBalancePaths (see page 621)	Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.
	GetLibraryItem (see page 622)	Loads the library item's data. This method is overloaded.
	GetLibrarySettings (see page 623)	Loads the library settings.
	GetLibraryTypes (see page 623)	Loads the library types.
	GetLoadBalancePath (see page 624)	Gets the load balance path by ID.
	InternalLibrarySearch (see page 624)	Searches library items.
	UpdateLibrary (see page 624)	Updates a library item.
	UpdateLibrarySettings (see page 625)	Update the library settings.
	UpdateLoadBalanceSettings (see page 625)	Updates the load balance settings.
	UpdateQlinkTemplates (see page 625)	Updates the Quicklink template.

Topics

Name	Description
Library Methods (see page 618)	The methods of the Library class are listed here.

Legend

	Method

Library Methods

	Name	Description
	AddLoadBalanceItem (see page 619)	Adds a load balance path for files and images.
	DeleteLibraryItem (see page 619)	Deletes the library item by extension.
	DeleteLibraryItemById (see page 619)	Deletes the library items from the system.
	DeleteLoadBalanceItem (see page 620)	Deletes the load balance item from the system.
	GetAllChildLibItems (see page 620)	Loads all of the library items by type.
	GetAllLoadBalancePaths (see page 621)	Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.
	GetLibraryItem (see page 622)	Loads the library item's data. This method is overloaded.
	GetLibrarySettings (see page 623)	Loads the library settings.
	GetLibraryTypes (see page 623)	Loads the library types.
	GetLoadBalancePath (see page 624)	Gets the load balance path by ID.
	InternalLibrarySearch (see page 624)	Searches library items.
	UpdateLibrary (see page 624)	Updates a library item.
	UpdateLibrarySettings (see page 625)	Update the library settings.
	UpdateLoadBalanceSettings (see page 625)	Updates the load balance settings.
	UpdateQlinkTemplates (see page 625)	Updates the Quicklink template.

Legend

	Method

Library Methods

The methods of the Library class are listed here.

Public Methods

	Name	Description
	AddLoadBalanceItem (see page 619)	Adds a load balance path for files and images.
	DeleteLibraryItem (see page 619)	Deletes the library item by extension.
	DeleteLibraryItemById (see page 619)	Deletes the library items from the system.
	DeleteLoadBalanceItem (see page 620)	Deletes the load balance item from the system.
	GetAllChildLibItems (see page 620)	Loads all of the library items by type.
	GetAllLoadBalancePaths (see page 621)	Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.
	GetLibraryItem (see page 622)	Loads the library item's data. This method is overloaded.
	GetLibrarySettings (see page 623)	Loads the library settings.
	GetLibraryTypes (see page 623)	Loads the library types.

	GetLoadBalancePath (see page 624)	Gets the load balance path by ID.
	InternalLibrarySearch (see page 624)	Searches library items.
	UpdateLibrary (see page 624)	Updates a library item.
	UpdateLibrarySettings (see page 625)	Update the library settings.
	UpdateLoadBalanceSettings (see page 625)	Updates the load balance settings.
	UpdateQlinkTemplates (see page 625)	Updates the Quicklink template.

Legend

	Method

Library.AddLoadBalanceItem Method

Adds a load balance path for files and images.

C#

```
public void AddLoadBalanceItem(string Type, Collection pagedata);
```

Visual Basic

```
Public Function AddLoadBalanceItem(Type As string, pagedata As Collection) As void
```

Parameters

Parameters	Description
Type	The path type, either images or files
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.

C#

```
public void DeleteLibraryItem(Collection pagedata);
```

Visual Basic

```
Public Function DeleteLibraryItem(pagedata As Collection) As void
```

Parameters

Parameters	Description
pagedata	Library (see page 616) 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.

C#

```
public void DeleteLibraryItemById(Collection pagedata);
```

Visual Basic

```
Public Function DeleteLibraryItemById(pagedata As Collection) As void
```

Parameters

Parameters	Description
pagedata	Library (see page 616) 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteLoadBalanceItem", Description = "")]
public void DeleteLoadBalanceItem(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteLoadBalanceItem", Description = "")>
Public Function DeleteLoadBalanceItem(Id As Integer) As void
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllChildLibItems", Description = "Loads all of the library items
by type")]
public LibraryData[] GetAllChildLibItems(string Type, int ParentId, string OrderBy, int
currentPageNum, int pageSize, ref int totalPages);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllChildLibItems", Description = "Loads all of the library items
by type")>
Public Function GetAllChildLibItems(Type As string, ParentId As Integer, OrderBy As string,
currentPageNum As Integer, pageSize As Integer, totalPages As ref int) As LibraryData[]
```

Parameters

Parameters	Description
Type	The type (Required)
ParentId	The folder's ID
OrderBy	The order to load the library items
currentPageNum	The current page number
pageSize	The page size
totalPages	The total pages

Returns

Ektron.Cms.LibraryData()

Remarks

User login required to perform this operation

GetAllLoadBalancePaths Method

Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.

Overload List

	Name	Description
	Library.GetAllLoadBalancePaths (int, string) (see page 621)	Returns an array of LoadBalanceData.
	Library.GetAllLoadBalancePaths (string) (see page 621)	Returns all the load balance paths as a array of LoadBalanceData.

Library.GetAllLoadBalancePaths Method (int, string)

Returns an array of LoadBalanceData.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllLoadBalancePaths By Id and Type", Description = "")]
public LoadBalanceData[] GetAllLoadBalancePaths(int Id, string Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllLoadBalancePaths By Id and Type", Description = "")>
Public Function GetAllLoadBalancePaths(Id As Integer, Type As string) As LoadBalanceData[]
```

Parameters

Parameters	Description
Id	Load balance ID
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllLoadBalancePaths By Type", Description = "")]
public LoadBalanceData[] GetAllLoadBalancePaths(string Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllLoadBalancePaths By Type", Description = "")>
Public Function GetAllLoadBalancePaths(Type As string) As LoadBalanceData[]
```

Parameters

Parameters	Description
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

Loads the library item's data. This method is overloaded.

Overload List

	Name	Description
	Library.GetLibraryItem (int) (see page 622)	Loads the library item data.
	Library.GetLibraryItem (int, int) (see page 622)	Loads the library item's data.

Library.GetLibraryItem Method (int)

Loads the library item data.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetLibraryItem By Id", Description = "")]
public LibraryData GetLibraryItem(int LibID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetLibraryItem By Id", Description = "")>
Public Function GetLibraryItem(LibID As Integer) As LibraryData
```

Parameters

Parameters	Description
LibID	The library ID

Returns

Ektron.Cms.LibraryData

Library.GetLibraryItem Method (int, int)

Loads the library item's data.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetLibraryItem By Id And ParentID", Description = "")]
public LibraryData GetLibraryItem(int Id, int ParentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetLibraryItem By Id And ParentID", Description = "")>
```

```
Public Function GetLibraryItem(Id As Integer, ParentId As Integer) As LibraryData
```

Parameters

Parameters	Description
Id	The library's ID
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetLibrarySettings", Description = "")]
public LibraryConfigData GetLibrarySettings(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetLibrarySettings", Description = "")>
Public Function GetLibrarySettings(Id As Integer) As LibraryConfigData
```

Parameters

Parameters	Description
Id	The folder's ID

Returns

Ektron.Cms.LibraryConfigData

Remarks

Login required to perform this operation.

Library.GetLibraryTypes Method

Loads the library types.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetLibraryTypes", Description = "")]
public LibraryTypeData[] GetLibraryTypes(int Id, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetLibraryTypes", Description = "")>
Public Function GetLibraryTypes(Id As Integer, OrderBy As string) As LibraryTypeData[]
```

Parameters

Parameters	Description
Id	The folder's ID
OrderBy	Order by clause

Returns

Object

Remarks

Library (see page 616) read permission is required to perform this operation.

Library.GetLoadBalancePath Method

Gets the load balance path by ID.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "", Description = "")]
public LoadBalanceData GetLoadBalancePath(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "", Description = "")>
Public Function GetLoadBalancePath(Id As Integer) As LoadBalanceData
```

Parameters

Parameters	Description
Id	th load balance path's ID

Returns

Ektron.Cms.LoadBalanceData

Library.InternalLibrarySearch Method

Searches library items.

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
pagedata	Search criteria
sar	Search asset criteria

Returns

Ektron.Cms.SearchResultData()

Remarks

User login is required to perform this operation.

Library.UpdateLibrary Method

Updates a library item.

C#

```
public void UpdateLibrary(Collection pagedata);
```

Visual Basic

```
Public Function UpdateLibrary(pagedata As Collection) As void
```

Parameters

Parameters	Description
pagedata	Library (see page 616) item data

Library.UpdateLibrarySettings Method

Update the library settings.

C#

```
public void UpdateLibrarySettings(Collection pagedata);
```

Visual Basic

```
Public Function UpdateLibrarySettings(pagedata As Collection) As void
```

Parameters

Parameters	Description
pagedata	Library (see page 616) settings data

Remarks

A user with administrator privileges must be logged in to perform this operation.

Library.UpdateLoadBalanceSettings Method

Updates the load balance settings.

C#

```
public void UpdateLoadBalanceSettings(string Id, Collection pagedata);
```

Visual Basic

```
Public Function UpdateLoadBalanceSettings(Id As string, pagedata As Collection) As void
```

Parameters

Parameters	Description
Id	The load balance data
pagedata	The load balance items

Description

[WebMethod(MessageName = "UpdateLoadBalanceSettings", Description = "")]

Remarks

User login with administrator privileges are required to perform this operation.

Library.UpdateQlinkTemplates Method

Updates the Quicklink template.

C#

```
public void UpdateQlinkTemplates(Collection pagedata);
```

Visual Basic

```
Public Function UpdateQlinkTemplates(pagedata As Collection) As void
```

Parameters

Parameters	Description
pagedata	Quicklink template data

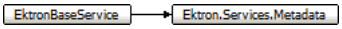
Remarks

A user with administrator privileges must be logged in to perform this operation.

Metadata Class

Summary description for Metadata

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

Members

Metadata Methods

	Name	Description
	AddMetaDataType (see page 627)	Adds a new metadata type to the system
	DeleteMetadataType (see page 627)	Deletes a metadata type from the system.
	GetContentMetadataList (see page 628)	
	GetMetadataType (see page 629)	Loads all content (see page 337) metadata.
	GetMetaDataTypes (see page 629)	Loads all the metadata types.
	UpdateMetadataType (see page 630)	Updates the metadata type.

Topics

Name	Description
Metadata Methods (see page 627)	The methods of the Metadata class are listed here.

Legend

	Method

Metadata Methods

	Name	Description
	AddMetaDataType (see page 627)	Adds a new metadata type to the system
	DeleteMetadataType (see page 627)	Deletes a metadata type from the system.
	GetContentMetadataList (see page 628)	
	GetMetadataType (see page 629)	Loads all content (see page 337) metadata.

	GetMetaDataTypes (see page 629)	Loads all the metadata types.
	UpdateMetadataType (see page 630)	Updates the metadata type.

Legend

	Method

Metadata Methods

The methods of the Metadata class are listed here.

Public Methods

	Name	Description
	AddMetaDataType (see page 627)	Adds a new metadata type to the system
	DeleteMetadataType (see page 627)	Deletes a metadata type from the system.
	GetContentMetadataList (see page 628)	
	GetMetadataType (see page 629)	Loads all content (see page 337) metadata.
	GetMetaDataTypes (see page 629)	Loads all the metadata types.
	UpdateMetadataType (see page 630)	Updates the metadata type.

Legend

	Method

Metadata.AddMetaDataType Method

Adds a new metadata type to the system

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddMetaDataType", Description = "Adds a new metadata type
definition")]
public void AddMetaDataType(ref ContentMetaData metaDataItem);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddMetaDataType", Description = "Adds a new metadata type
definition")>
Public Function AddMetaDataType(metaDataItem As ref ContentMetaData) As void
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
```

```
[WebMethod(MessageName = "DeleteMetadataType", Description = "")]
public void DeleteMetadataType(int MetadataId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteMetadataType", Description = "")>
Public Function DeleteMetadataType(MetadataId As Integer) As void
```

Parameters

Parameters	Description
MetadataId	Metatype ID

Remarks

A user with administrator privileges must be logged in to perform this operation.

GetContentMetadataList Method

Overload List

	Name	Description
	Metadata.GetContentMetadataList (int) (see page 628)	Loads all of the metadata for the given content (see page 337).
	Metadata.GetContentMetadataList (int, bool) (see page 628)	Loads all of the metadata for the given content (see page 337).

Metadata.GetContentMetadataList Method (int)

Loads all of the metadata for the given content (see page 337).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetContentMetadataList", Description = "")]
public CustomAttributeList GetContentMetadataList(int ContentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetContentMetadataList", Description = "")>
Public Function GetContentMetadataList(ContentId As Integer) As CustomAttributeList
```

Parameters

Parameters	Description
ContentId	The contents ID.

Returns

Ektron.Cms.CustomAttributeList

Remarks

Read permissions required.

Metadata.GetContentMetadataList Method (int, bool)

Loads all of the metadata for the given content (see page 337).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetContentMetadataList with preview", Description = "")]
public CustomAttributeList GetContentMetadataList(int ContentId, bool preview);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetContentMetadataList with preview", Description = "")>
Public Function GetContentMetadataList(ContentId As Integer, preview As bool) As
CustomAttributeList
```

Parameters

Parameters	Description
ContentId	The contents ID.
preview	True implies get metadata from staging content (see page 337), else use published content (see page 337)

Returns

Ektron.Cms.CustomAttributeList

Remarks

Read permissions required.

Metadata.GetMetadataType Method

Loads all content (see page 337) metadata.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetMetadataType", Description = "")]
public ContentMetaData GetMetadataType(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetMetadataType", Description = "")>
Public Function GetMetadataType(Id As Integer) As ContentMetaData
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetMetaDataTypes", Description = "")]
public ContentMetaData[] GetMetaDataTypes(string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetMetaDataTypes", Description = "")>
Public Function GetMetaDataTypes(OrderBy As string) As ContentMetaData[]
```

Parameters

Parameters	Description
OrderBy	Order by clause

Returns

Ektron.Cms.ContentMetaData()

Remarks

User 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
pagedata	Metadata (see page 626) types

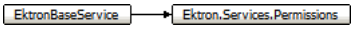
Remarks

A user with administrator privileges must be logged in to perform this operation.

Permissions Class

Summary description for Permissions

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

Members

Permissions Methods

	Name	Description
	AddItemPermission (see page 632)	Adds the item permissions for a user or a usergroup.
	DeleteItemPermission (see page 632)	Deletes a user or a group from the item's permissions

	DisableContentInheritance (see page 633)	Disable permissions inheritance on the content (see page 337) and copies current permissions to the content (see page 337).
	DisableFolderInheritance (see page 633)	Disable permissions inheritance on the folder and copies current permissions to the folder.
	EnableContentInheritance (see page 633)	Deletes current permissions from the content (see page 337) and enable permissions inheritance on the content (see page 337).
	EnableFolderInheritance (see page 634)	Deletes current permissions from the folder and enable permissions inheritance on the folder.
	GetUserPermissions (see page 634)	
	LoadPermissions (see page 636)	Loads the permission for given object, such as content (see page 337), folder, task, etc.
	UpdateItemPermission (see page 636)	Updates the item permissions for a user or a usergroup.

Topics

Name	Description
Permissions Methods (see page 631)	The methods of the Permissions class are listed here.

Legend

	Method

Permissions Methods

	Name	Description
	AddItemPermission (see page 632)	Adds the item permissions for a user or a usergroup.
	DeleteItemPermission (see page 632)	Deletes a user or a group from the item's permissions
	DisableContentInheritance (see page 633)	Disable permissions inheritance on the content (see page 337) and copies current permissions to the content (see page 337).
	DisableFolderInheritance (see page 633)	Disable permissions inheritance on the folder and copies current permissions to the folder.
	EnableContentInheritance (see page 633)	Deletes current permissions from the content (see page 337) and enable permissions inheritance on the content (see page 337).
	EnableFolderInheritance (see page 634)	Deletes current permissions from the folder and enable permissions inheritance on the folder.
	GetUserPermissions (see page 634)	
	LoadPermissions (see page 636)	Loads the permission for given object, such as content (see page 337), folder, task, etc.
	UpdateItemPermission (see page 636)	Updates the item permissions for a user or a usergroup.

Legend

	Method

Permissions Methods

The methods of the Permissions class are listed here.

Public Methods

	Name	Description
	AddItemPermission (see page 632)	Adds the item permissions for a user or a usergroup.
	DeleteItemPermission (see page 632)	Deletes a user or a group from the item's permissions
	DisableContentInheritance (see page 633)	Disable permissions inheritance on the content (see page 337) and copies current permissions to the content (see page 337).
	DisableFolderInheritance (see page 633)	Disable permissions inheritance on the folder and copies current permissions to the folder.

	EnableContentInheritance (see page 633)	Deletes current permissions from the content (see page 337) and enable permissions inheritance on the content (see page 337).
	EnableFolderInheritance (see page 634)	Deletes current permissions from the folder and enable permissions inheritance on the folder.
	GetUserPermissions (see page 634)	
	LoadPermissions (see page 636)	Loads the permission for given object, such as content (see page 337), folder, task, etc.
	UpdateItemPermission (see page 636)	Updates the item permissions for a user or a usergroup.

Legend

	Method

Permissions.AddItemPermission Method

Adds the item permissions for a user or a usergroup.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddItemPermission", Description = "Adds the item permissions for a
user or a usergroup.")]
public bool AddItemPermission(UserPermissionData userPermData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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
userPermData	User's Permissions (see page 630) Data

Returns

False if it's successful

Permissions.DeleteItemPermission Method

Deletes a user or a group from the item's permissions

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteItemPermission", Description = "Deletes a user or a group from
the item's permissions")]
public bool DeleteItemPermission(int id, bool IsGroupId, int itemId,
EkEnumeration.CMSObjectTypes itemType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteItemPermission", Description = "Deletes a user or a group from
the item's permissions")>
Public Function DeleteItemPermission(id As Integer, IsGroupId As bool, itemId As Integer,
itemType As EkEnumeration.CMSObjectTypes) As bool
```

Parameters

Parameters	Description
id	User or Group Id
IsGroupId	Given ID is a user id or a group id?
itemId	ContentId or FolderId

itemType	Given item id is EkEnumeration.CMSObjectTypes.Content (see page 576) or EkEnumeration.CMSObjectTypes.Folder?

Returns

False if it's successful.

Permissions.DisableContentInheritance Method

Disable permissions inheritance on the content (see page 337) and copies current permissions to the content (see page 337).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DisableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool DisableContentInheritance(int id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DisableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")>
Public Function DisableContentInheritance(id As Integer) As bool
```

Parameters

Parameters	Description
id	Content (see page 576) Id

Returns

False if it's successful.

Permissions.DisableFolderInheritance Method

Disable permissions inheritance on the folder and copies current permissions to the folder.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DisableFolderInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool DisableFolderInheritance(int id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DisableFolderInheritance", Description = "Deletes a user or a group
from the item's permissions")>
Public Function DisableFolderInheritance(id As Integer) As bool
```

Parameters

Parameters	Description
id	Folder Id

Returns

False if it's successful.

Permissions.EnableContentInheritance Method

Deletes current permissions from the content (see page 337) and enable permissions inheritance on the content (see page 337).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "EnableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool EnableContentInheritance(int id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "EnableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")>
Public Function EnableContentInheritance(id As Integer) As bool
```

Parameters

Parameters	Description
id	Content (see page 576) Id

Returns

False if it's successful.

Permissions.EnableFolderInheritance Method

Deletes current permissions from the folder and enable permissions inheritance on the folder.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "EnableFolderInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool EnableFolderInheritance(int id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "EnableFolderInheritance", Description = "Deletes a user or a group
from the item's permissions")>
Public Function EnableFolderInheritance(id As Integer) As bool
```

Parameters

Parameters	Description
id	Folder Id

Returns

False if it's successful.

GetUserPermissions Method

Overload List

	Name	Description
	Permissions.GetUserPermissions (int, string, int, string) (see page 635)	Loads the user permissions as a array of UserPermissionData.
	Permissions.GetUserPermissions (int, string, int, string, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) (see page 635)	Loads the user permissions as a array of UserPermissionData.

Permissions.GetUserPermissions Method (int, string, int, string)

Loads the user permissions as a array of UserPermissionData.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUserPermissions", Description = "Loads the user permissions as a
array of UserPermissionData.")]
public UserPermissionData[] GetUserPermissions(int Id, string ItemType, int UserId, string
UserGroupList);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUserPermissions", Description = "Loads the user permissions as a
array of UserPermissionData.")>
Public Function GetUserPermissions(Id As Integer, ItemType As string, UserId As Integer,
UserGroupList As string) As UserPermissionData[]
```

Parameters

Parameters	Description
Id	the item's ID
ItemType	The item's type
UserId	The user's ID
UserGroupList	the group list

Returns

Ektron.Cms.UserPermissionData()

Permissions.GetUserPermissions Method (int, string, int, string, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType)

Loads the user permissions as a array of UserPermissionData.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUserPermissions By PermissionType", Description = "Loads the user
permissions as a array of UserPermissionData.")]
public UserPermissionData[] GetUserPermissions(int Id, string ItemType, int UserId, string
UserGroupList, ContentAPI.PermissionUserType PermissionType, ContentAPI.PermissionRequestType
PermissionRequest);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUserPermissions By PermissionType", Description = "Loads the user
permissions as a array of UserPermissionData.")>
Public Function GetUserPermissions(Id As Integer, ItemType As string, UserId As Integer,
UserGroupList As string, PermissionType As ContentAPI.PermissionUserType, PermissionRequest As
ContentAPI.PermissionRequestType) As UserPermissionData[]
```

Parameters

Parameters	Description
Id	the item's ID
ItemType	The item's type
UserId	The user's ID
UserGroupList	the group list
PermissionType	PermissionUserType
PermissionRequest	PermissionRequestType

Returns

Ektron.Cms.UserPermissionData()

Permissions.LoadPermissions Method

Loads the permission for given object, such as content (see page 337), folder, task, etc.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "LoadPermissions", Description = "Loads the permission for given
object, such as content, folder, task, etc.")]
public PermissionData LoadPermissions(int Id, string RequestType,
ContentAPI.PermissionResultType Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "LoadPermissions", Description = "Loads the permission for given
object, such as content, folder, task, etc.")>
Public Function LoadPermissions(Id As Integer, RequestType As string, Type As
ContentAPI.PermissionResultType) As PermissionData
```

Parameters

Parameters	Description
Id	The object ID
RequestType	The object type := content (see page 337) | folder | collections | tasks
Type	The permission result type

Returns

Ektron.Cms.PermissionData

Permissions.UpdateItemPermission Method

Updates the item permissions for a user or a usergroup.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateItemPermission", Description = "Updates the item permissions
for a user or a usergroup.")]
public bool UpdateItemPermission(UserPermissionData userPermData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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
userPermData	User's Permissions (see page 630) Data

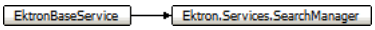
Returns

False if it's successful

SearchManager Class

Summary description for SearchManager

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

Members

SearchManager Methods

	Name	Description
	ExecuteSearch (see page 637)	Returns the array of ContentData. This method is overloaded.
	Search (see page 639)	

Topics

Name	Description
SearchManager Methods (see page 637)	The methods of the SearchManager class are listed here.

Legend

	Method

SearchManager Methods

	Name	Description
	ExecuteSearch (see page 637)	Returns the array of ContentData. This method is overloaded.
	Search (see page 639)	

Legend

	Method

SearchManager Methods

The methods of the SearchManager class are listed here.

Public Methods

	Name	Description
	ExecuteSearch (see page 637)	Returns the array of ContentData. This method is overloaded.
	Search (see page 639)	

Legend

	Method

ExecuteSearch Method

Returns the array of ContentData. This method is overloaded.

Overload List

	Name	Description
	SearchManager.ExecuteSearch (ContentSearchCondition) (see page 638)	Returns the array of ContentData based on the criteria ContentSearchCondition.
	SearchManager.ExecuteSearch (UserSearchCondition) (see page 638)	Returns the array of UserData based on the criteria UserSearchCondition.

SearchManager.ExecuteSearch Method (ContentSearchCondition)

Returns the array of ContentData based on the criteria ContentSearchCondition.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "ExecuteSearch For Content", Description = "")]
public ContentData[] ExecuteSearch(ContentSearchCondition condition);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "ExecuteSearch For Content", Description = "")>
Public Function ExecuteSearch(condition As ContentSearchCondition) As ContentData[]
```

Parameters

Parameters	Description
condition	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.ExecuteSearch Method (UserSearchCondition)

Returns the array of UserData based on the criteria UserSearchCondition.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "ExecuteSearch For User", Description = "")]
```

```
public UserData[] ExecuteSearch(UserSearchCondition condition);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "ExecuteSearch For User", Description = "")>
Public Function ExecuteSearch(condition As UserSearchCondition) As UserData[]
```

Parameters

Parameters	Description
condition	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

Overload List

	Name	Description
	SearchManager.Search (SearchRequestData, ref int) (see page 639)	Complete this API
	SearchManager.Search (string, ref int) (see page 640)	Complete this API

SearchManager.Search Method (SearchRequestData, ref int)

Complete this API

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "ExecuteSearch search complete", Description = "")]
public SearchResponseData[] Search(SearchRequestData searchReqData, ref int resultCount);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "ExecuteSearch search complete", Description = "")>
```

```
Public Function Search(searchReqData As SearchRequestData, resultCount As ref int) As
SearchResponseData[]
```

Parameters

Parameters	Description
searchReqData	
resultCount	

SearchManager.Search Method (string, ref int)

Complete this API

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "ExecuteSearch basic", Description = "")]
public SearchResponseData[] Search(string searchText, ref int resultCount);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "ExecuteSearch basic", Description = "")>
Public Function Search(searchText As string, resultCount As ref int) As SearchResponseData[]
```

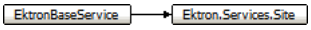
Parameters

Parameters	Description
resultCount	
searchReqData	

Site Class

Summary description for Site

Class Hierarchy



C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class Site : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class Site
Inherits EktronBaseService
```

File

Site.cs

Members

Site Methods

	Name	Description
	BreakTaskPermissionInherit (see page 642)	Removes the task permissions for a given user.
	GetAllActiveLanguages (see page 643)	Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language Settings area.
	GetAllLanguages (see page 643)	Gets a list of all the available languages.

	GetClientDayNames (see page 643)	Returns the day name of a given content (see page 337) language (or site language).
	GetClientMonthNames (see page 644)	Returns the month name of a given content (see page 337) language (or site language)
	GetCookieValue (see page 644)	Reads the cookie value.
	GetLanguage (see page 644)	Returns the details about the given language ID.
	GetLanguages (see page 645)	Gets a list of all languages that meet a given state, for example, active.
	GetSiteVariables (see page 645)	Returns all of the settings data.
	GetTaskPermission (see page 645)	Returns 1 if the user has one of the task permissions, otherwise 0.
	IsTaskPermissionInherited (see page 646)	Returns true if a user has already granted one of the task permissions.
	SetCookieValue (see page 646)	Resets the cookie value.
	SetTaskPermission (see page 647)	Updates the task permission(s) for the given user or group.
	UpdateSiteEnabledLanguages (see page 647)	Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled.

Topics

Name	Description
Site Methods (see page 642)	The methods of the Site class are listed here.

Legend

	Method

Site Methods

	Name	Description
	BreakTaskPermissionInherit (see page 642)	Removes the task permissions for a given user.
	GetAllActiveLanguages (see page 643)	Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language Settings area.
	GetAllLanguages (see page 643)	Gets a list of all the available languages.
	GetClientDayNames (see page 643)	Returns the day name of a given content (see page 337) language (or site language).
	GetClientMonthNames (see page 644)	Returns the month name of a given content (see page 337) language (or site language)
	GetCookieValue (see page 644)	Reads the cookie value.
	GetLanguage (see page 644)	Returns the details about the given language ID.
	GetLanguages (see page 645)	Gets a list of all languages that meet a given state, for example, active.
	GetSiteVariables (see page 645)	Returns all of the settings data.
	GetTaskPermission (see page 645)	Returns 1 if the user has one of the task permissions, otherwise 0.
	IsTaskPermissionInherited (see page 646)	Returns true if a user has already granted one of the task permissions.
	SetCookieValue (see page 646)	Resets the cookie value.
	SetTaskPermission (see page 647)	Updates the task permission(s) for the given user or group.
	UpdateSiteEnabledLanguages (see page 647)	Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled.

Legend

	Method

Site Methods

The methods of the Site class are listed here.

Public Methods

	Name	Description
	BreakTaskPermissionInherit (see page 642)	Removes the task permissions for a given user.
	GetAllActiveLanguages (see page 643)	Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language Settings area.
	GetAllLanguages (see page 643)	Gets a list of all the available languages.
	GetClientDayNames (see page 643)	Returns the day name of a given content (see page 337) language (or site language).
	GetClientMonthNames (see page 644)	Returns the month name of a given content (see page 337) language (or site language)
	GetCookieValue (see page 644)	Reads the cookie value.
	GetLanguage (see page 644)	Returns the details about the given language ID.
	GetLanguages (see page 645)	Gets a list of all languages that meet a given state, for example, active.
	GetSiteVariables (see page 645)	Returns all of the settings data.
	GetTaskPermission (see page 645)	Returns 1 if the user has one of the task permissions, otherwise 0.
	IsTaskPermissionInherited (see page 646)	Returns true if a user has already granted one of the task permissions.
	SetCookieValue (see page 646)	Resets the cookie value.
	SetTaskPermission (see page 647)	Updates the task permission(s) for the given user or group.
	UpdateSiteEnabledLanguages (see page 647)	Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled.

Legend

	Method

Site.BreakTaskPermissionInherit Method

Removes the task permissions for a given user.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "BreakTaskPermissionInherit", Description = "")]
public void BreakTaskPermissionInherit(int UserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "BreakTaskPermissionInherit", Description = "")>
Public Function BreakTaskPermissionInherit(UserId As Integer) As void
```

Parameters

Parameters	Description
UserId	User 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllActiveLanguages", Description = "")]
public LanguageData[] GetAllActiveLanguages();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllActiveLanguages", Description = "")>
Public Function GetAllActiveLanguages() As LanguageData[]
```

Returns

Array of Ektron.Cms.LanguageData

Site.GetAllLanguages Method

Gets a list of all the available languages.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllLanguages", Description = "")]
public LanguageData[] GetAllLanguages(string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllLanguages", Description = "")>
Public Function GetAllLanguages(OrderBy As string) As LanguageData[]
```

Parameters

Parameters	Description
OrderBy	Valid values are language_id, language_name or any field name from the language_type table.

Returns

Array of Ektron.Cms.LanguageData

Site.GetClientDayNames Method

Returns the day name of a given content (see page 337) language (or site language).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetClientDayNames", Description = "")]
public string GetClientDayNames();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetClientDayNames", Description = "")>
Public Function GetClientDayNames() As string
```

Returns

String

Site.GetClientMonthNames Method

Returns the month name of a given content (see page 337) language (or site language)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetClientMonthNames", Description = "")]
public string GetClientMonthNames();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetClientMonthNames", Description = "")>
Public Function GetClientMonthNames() As string
```

Returns

String

Site.GetCookieValue Method

Reads the cookie value.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetCookieValue", Description = "")]
public string GetCookieValue(string Keyname);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetCookieValue", Description = "")>
Public Function GetCookieValue(Keyname As string) As string
```

Parameters

Parameters	Description
Keyname	The Keyname. (Required)

Returns

String

Site.GetLanguage Method

Returns the details about the given language ID.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetLanguage", Description = "")]
public LanguageData GetLanguage(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetLanguage", Description = "")>
Public Function GetLanguage(Id As Integer) As LanguageData
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetLanguages", Description = "")]
public LanguageDataSet.LanguageDataTable GetLanguages(LanguageDataSet.LanguageState State,
LanguageDataSet.SortBy SortBy, LanguageDataSet.SortDirection SortDirection);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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
State	The state that the language must meet. For example, defined, active, site enabled.
SortBy	The field on which to sort the languages in the data table.
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetSiteVariables", Description = "")]
public SettingsData GetSiteVariables(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetSiteVariables", Description = "")>
Public Function GetSiteVariables(Id As Integer) As SettingsData
```

Parameters

Parameters	Description
Id	User 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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetTaskPermission", Description = "")]
public int GetTaskPermission(int Id, int Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTaskPermission", Description = "")>
Public Function GetTaskPermission(Id As Integer, Type As Integer) As Integer
```

Parameters

Parameters	Description
Id	User or group ID
Type	User or group (1 = user, 2 = group).

Returns

Integer

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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "IsTaskPermissionInherited", Description = "")]
public bool IsTaskPermissionInherited(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "IsTaskPermissionInherited", Description = "")>
Public Function IsTaskPermissionInherited(Id As Integer) As bool
```

Parameters

Parameters	Description
Id	User 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "SetCookieValue", Description = "")]
public void SetCookieValue(string Keyname, string CookieValue);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "SetCookieValue", Description = "")>
Public Function SetCookieValue(Keyname As string, CookieValue As string) As void
```

Parameters

Parameters	Description
Keyname	The Keyname. (Required)

CookieValue	Value

Site.SetTaskPermission Method

Updates the task permission(s) for the given user or group.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "SetTaskPermission", Description = "")]
public void SetTaskPermission(int Id, int UserType, int PermissionType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "SetTaskPermission", Description = "")>
Public Function SetTaskPermission(Id As Integer, UserType As Integer, PermissionType As
Integer) As void
```

Parameters

Parameters	Description
Id	User ID or Group ID
UserType	User (1) or Group (2)
PermissionType	Ektron.Cms.Common.TaskPermission (1 = Create Task, 2 = Redirect Task, 3 = Delete Task, 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateSiteEnabledLanguages", Description = "")]
public void UpdateSiteEnabledLanguages(string LanguageIDs);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UpdateSiteEnabledLanguages", Description = "")>
Public Function UpdateSiteEnabledLanguages(LanguageIDs As string) As void
```

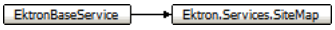
Parameters

Parameters	Description
LanguageIDs	Comma-separated list of language ids. For example, "1033,1031"

SiteMap Class

Summary description for SiteMap

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

Members

SiteMap Methods

	Name	Description
	GetFolderBreadcrumbPath (see page 649)	Get breadcrumb path of the folder in which content (see page 337) block exists.
	GetWebsiteMap (see page 649)	Creates the site map from folder breadcrumb.
	RenderFolderBreadcrumbHtml (see page 649)	Render folder breadcrumb HTML
	RenderWebsiteMap (see page 650)	Create HTML to render on the page.

Topics

Name	Description
SiteMap Methods (see page 648)	The methods of the SiteMap class are listed here.

Legend

	Method

SiteMap Methods

	Name	Description
	GetFolderBreadcrumbPath (see page 649)	Get breadcrumb path of the folder in which content (see page 337) block exists.
	GetWebsiteMap (see page 649)	Creates the site map from folder breadcrumb.
	RenderFolderBreadcrumbHtml (see page 649)	Render folder breadcrumb HTML
	RenderWebsiteMap (see page 650)	Create HTML to render on the page.

Legend

	Method

SiteMap Methods

The methods of the SiteMap class are listed here.

Public Methods

	Name	Description
	GetFolderBreadcrumbPath (see page 649)	Get breadcrumb path of the folder in which content (see page 337) block exists.
	GetWebsiteMap (see page 649)	Creates the site map from folder breadcrumb.
	RenderFolderBreadcrumbHtml (see page 649)	Render folder breadcrumb HTML
	RenderWebsiteMap (see page 650)	Create HTML to render on the page.

Legend

	Method

SiteMap.GetFolderBreadcrumbPath Method

Get breadcrumb path of the folder in which content (see page 337) block exists.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetFolderBreadcrumbPath", Description = "")]
public SitemapPath[] GetFolderBreadcrumbPath(int id, bool isFolder);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetFolderBreadcrumbPath", Description = "")>
Public Function GetFolderBreadcrumbPath(id As Integer, isFolder As bool) As SitemapPath[]
```

Parameters

Parameters	Description
id	Content (see page 576) or Folder Id
isFolder	set it true if the id is folder id

Returns

Array of node in the path

SiteMap.GetWebsiteMap Method

Creates the site map from folder breadcrumb.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetWebsiteMap", Description = "")]
public WebsiteMap GetWebsiteMap(int folderId, bool getSameLevelNodes);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetWebsiteMap", Description = "")>
Public Function GetWebsiteMap(folderId As Integer, getSameLevelNodes As bool) As WebsiteMap
```

Parameters

Parameters	Description
folderId	Build the web sitemap from the folder specified.
getSameLevelNodes	True indicates include all sites at same level as folderId

SiteMap.RenderFolderBreadcrumbHtml Method

Render folder breadcrumb HTML

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "RenderFolderBreadcrumbHtml", Description = "")]
public string RenderFolderBreadcrumbHtml(SitemapPath[] nodes, string pathSeparator, bool
linkNodes, string linkTarget, bool displayVertical, string pageUrl);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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
nodes	
pathSeparator	
linkNodes	
linkTarget	
displayVerticle	

Returns

HTML

SiteMap.RenderWebsiteMap Method

Create HTML to render on the page.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "RenderWebsiteMap", Description = "")]
public string RenderWebsiteMap(WebsiteMap webSitemap, string displayType, string className,
int maxLevel, int startLevel, string pageUrl);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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	GetWebSiteMap()
displayType	Determines how to produce HTML.
className	Stylesheet class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.
maxLevel	Set the max level of the sitemap. Set it to 0 for unlimited.
startLevel	Set the starting level of the sitemap. Set it to 0 for start from root.

Returns

HTML/XML

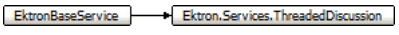
Remarks

Supported displayType: ListWithDescription | List | RawXml

ThreadedDiscussion Class

Summary description for ThreadedDiscussion

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

Members

ThreadedDiscussion Methods

	Name	Description
	AddBoard (see page 653)	Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns integer ID of new discussion board.
	AddForum (see page 653)	Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns integer ID of new discussion forum.
	AddTopic (see page 654)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer. This method is overloaded.
	DeleteBoard (see page 655)	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.
	DeleteForum (see page 655)	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.
	DeleteTopic (see page 655)	Deletes a topic by topicID This method will delete all posts associated with the topic as well. Returns boolean value.
	GetBoardCategories (see page 656)	Method to get the Categories for a board by ID Returns either zero length or populated array. See DiscussionCategory class for more information.
	GetBoardForums (see page 656)	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 657)	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.
	GetDiscussionBoardString (see page 658)	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.
	GetForum (see page 658)	Method to obtain a particular forum by ID. Will return a discussion board with a 1 length Discussion Forum array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.
	GetRepliesForTopics (see page 658)	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 659)	Method to obtain a particular Topic by ID. Will return a discussion board with a 1 length Discussion Forum 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 659)	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.

Topics

Name	Description
ThreadedDiscussion Methods (see page 652)	The methods of the ThreadedDiscussion class are listed here.

Legend

	Method

ThreadedDiscussion Methods

	Name	Description
	AddBoard (see page 653)	Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns integer ID of new discussion board.
	AddForum (see page 653)	Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns integer ID of new discussion forum.
	AddTopic (see page 654)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer. This method is overloaded.

	DeleteBoard (see page 655)	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.
	DeleteForum (see page 655)	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.
	DeleteTopic (see page 655)	Deletes a topic by topicID This method will delete all posts associated with the topic as well. Returns boolean value.
	GetBoardCategories (see page 656)	Method to get the Categories for a board by ID Returns either zero length or populated array. See DiscussionCategory class for more information.
	GetBoardForums (see page 656)	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 657)	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.
	GetDiscussionBoardString (see page 658)	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.
	GetForum (see page 658)	Method to obtain a particular forum by ID. Will return a discussion board with a 1 length Discussion Forum array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.
	GetRepliesForTopics (see page 658)	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 659)	Method to obtain a particular Topic by ID. Will return a discussion board with a 1 length Discussion Forum 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 659)	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.

Legend

	Method

ThreadedDiscussion Methods

The methods of the ThreadedDiscussion class are listed here.

Public Methods

	Name	Description
	AddBoard (see page 653)	Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns integer ID of new discussion board.
	AddForum (see page 653)	Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns integer ID of new discussion forum.
	AddTopic (see page 654)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer. This method is overloaded.
	DeleteBoard (see page 655)	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.
	DeleteForum (see page 655)	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.
	DeleteTopic (see page 655)	Deletes a topic by topicID This method will delete all posts associated with the topic as well. Returns boolean value.
	GetBoardCategories (see page 656)	Method to get the Categories for a board by ID Returns either zero length or populated array. See DiscussionCategory class for more information.
	GetBoardForums (see page 656)	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 657)	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.
	GetDiscussionBoardString (see page 658)	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.
	GetForum (see page 658)	Method to obtain a particular forum by ID. Will return a discussion board with a 1 length Discussion Forum array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.
	GetRepliesForTopics (see page 658)	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 659)	Method to obtain a particular Topic by ID. Will return a discussion board with a 1 length Discussion Forum 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 659)	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.

Legend

	Method

ThreadedDiscussion.AddBoard Method

Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns integer ID of new discussion board.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddBoard", Description = "")]
public int AddBoard(int ParentID, string Name, string Title, bool RequiresAuthentication,
string StyleSheet, string[] Categories);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddBoard", Description = "")>
Public Function AddBoard(ParentID As Integer, Name As string, Title As string,
RequiresAuthentication As bool, StyleSheet As string, Categories As string[]) As Integer
```

Parameters

Parameters	Description
ParentID	Integer ID of folder to add discussion board to.
Name	Name of board.
Title	Title of board.
RequiresAuthentication	Does the Board require authentication?
StyleSheet	The stylesheet to apply to the board.
Categories	String array of categories for the board.

Returns

Integer ID of new discussion board.

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 integer ID of new discussion forum.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddForum", Description = "")]
public int AddForum(int BoardID, string Name, string Description, bool ModeratePosts, bool
LockForum, int SortOrder, int CategoryID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddForum", Description = "")>
Public Function AddForum(BoardID As Integer, Name As string, Description As string,
ModeratePosts As bool, LockForum As bool, SortOrder As Integer, CategoryID As Integer) As
Integer
```

Parameters

Parameters	Description
BoardID	
Name	
Description	String value for the forum summary

ModeratePosts	Are posts moderated?
LockForum	Is the forum locked?
SortOrder	Sort order for the new forum.
CategoryID	Integer of the Category ID it belongs to.

Returns

Integer ID of new discussion forum.

AddTopic Method

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer. This method is overloaded.

Overload List

	Name	Description
	ThreadedDiscussion.AddTopic (int, string, string) (see page 654)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer.
	ThreadedDiscussion.AddTopic (int, string, string, int) (see page 654)	Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as integer.

ThreadedDiscussion.AddTopic Method (int, string, string)

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddTopic", Description = "")]
public int AddTopic(int ForumID, string Title, string Message);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddTopic", Description = "")>
Public Function AddTopic(ForumID As Integer, Title As string, Message As string) As Integer
```

Parameters

Parameters	Description
ForumID	Integer value of forum to add topic to.
Title	String value of the topic title.
Message	String value of the topic message.

Returns

Integer value for the topic ID.

ThreadedDiscussion.AddTopic Method (int, string, string, int)

Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as integer.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddTopic With Language ID", Description = "")]
public int AddTopic(int ForumID, string Title, string Message, int LanguageId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddTopic With Language ID", Description = "")>
Public Function AddTopic(ForumID As Integer, Title As string, Message As string, LanguageId As Integer) As Integer
```

Parameters

Parameters	Description
ForumID	Integer value of forum to add topic to.
Title	String value of the topic title.
Message	String value of the topic message.
LanguageId	Integer id of the Language ID

Returns

Integer value for the topic ID.

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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteBoard", Description = "")]
public void DeleteBoard(int BoardID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteBoard", Description = "")>
Public Function DeleteBoard(BoardID As Integer) As void
```

Parameters

Parameters	Description
BoardID	The integer 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteForum", Description = "")]
public void DeleteForum(int forumID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteForum", Description = "")>
Public Function DeleteForum(forumID As Integer) As void
```

Parameters

Parameters	Description
forumID	The integer ID of the forum. This corresponds to the folder id.

ThreadedDiscussion.DeleteTopic Method

Deletes a topic by topicID This method will delete all posts associated with the topic as well. Returns boolean value.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteTopic", Description = "")]
public bool DeleteTopic(int topicid);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteTopic", Description = "")>
Public Function DeleteTopic(topicid As Integer) As bool
```

Parameters

Parameters	Description
post_id	Integer 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 class for more information.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBoardCategories", Description = "")]
public DiscussionCategory[] GetBoardCategories(int boardID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetBoardCategories", Description = "")>
Public Function GetBoardCategories(boardID As Integer) As DiscussionCategory[]
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.

Returns

Returns an array of type DiscussionCategory.

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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetBoardForums", Description = "")]
public DiscussionForum[] GetBoardForums(int boardID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetBoardForums", Description = "")>
Public Function GetBoardForums(boardID As Integer) As DiscussionForum[]
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.

Returns

Returns an array of type DiscussionForum.

GetDiscussionBoard Method

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 (int) (see page 657)	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 657)	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.

ThreadedDiscussion.GetDiscussionBoard Method (int)

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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetDiscussionBoard By Board ID", Description = "")]
public DiscussionBoard GetDiscussionBoard(int boardID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetDiscussionBoard By Board ID", Description = "")>
Public Function GetDiscussionBoard(boardID As Integer) As DiscussionBoard
```

Parameters

Parameters	Description
boardID	Integer 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetDiscussionBoard By Board Path", Description = "")]
public DiscussionBoard GetDiscussionBoard(string boardpath);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetDiscussionBoard By Board Path", Description = "")>
Public Function GetDiscussionBoard(boardpath As string) As DiscussionBoard
```

Parameters

Parameters	Description
boardpath	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 CSS stylesheets.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "", Description = "")]
public string GetDiscussionBoardString(int boardID, string URLpath);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "", Description = "")>
Public Function GetDiscussionBoardString(boardID As Integer, URLpath As string) As string
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.
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 array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetForum", Description = "")]
public DiscussionBoard GetForum(int forumid);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetForum", Description = "")>
Public Function GetForum(forumid As Integer) As DiscussionBoard
```

Parameters

Parameters	Description
forumid	Integer 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

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetRepliesForTopics", Description = "")]
public ArrayList GetRepliesForTopics(int ForumID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetRepliesForTopics", Description = "")>
Public Function GetRepliesForTopics(ForumID As Integer) As ArrayList
```

Parameters

Parameters	Description
ForumID	Integer 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 658)(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 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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetTopic", Description = "")]
public DiscussionBoard GetTopic(int topicid);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetTopic", Description = "")>
Public Function GetTopic(topicid As Integer) As DiscussionBoard
```

Parameters

Parameters	Description
topicid	Integer 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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "SearchReplies", Description = "")]
public EkTasks SearchReplies(int[] ForumIDs, string[] SearchTerms, EkEnumeration.SearchTypes
stType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
```

```
<WebMethod(MessageName = "SearchReplies", Description = "")>
Public Function SearchReplies(ForumIDs As int[], SearchTerms As string[], stType As
EkEnumeration.SearchTypes) As EkTasks
```

Parameters

Parameters	Description
ForumIDs	Integer array of the Forum IDs you want to search. Use zero length array, or 0, to search all.
SearchTerms	String array of the search terms (case insensitive).
stType	EkEnumeration.SearchTypes enumeration value for the search type.

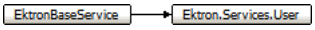
Returns

An instance of type ekTasks (an array of Tasks).

User Class

Summary description for User

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

Members

User Methods

	Name	Description
	ActivateUserAccount (see page 665)	Activates a deleted (deactivated) user into the current/active state using the UserId or the Username and GUID (AccountID) parameters. This method is overloaded.
	AddMembershipUser (see page 666)	Creates new membership user.
	AddNewUser (see page 666)	Creates new membership user.
	AddUser (see page 667)	Creates new cms user
	AddUserGroup (see page 667)	Creates new user group.
	AddUserToGroup (see page 668)	Adds an existing user to an existing group.
	AutoLogInUser (see page 668)	
	CreateBlankUserObject (see page 668)	This method is overloaded.
	DeleteUserByID (see page 669)	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 670)	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 670)	Removes a user from the designated group.

	DeleteUserGroup (see page 671)	Removes the group from the system permanently.
	EditUserCustomProperties (see page 671)	This method creates the editable fields(UI) for user's custom properties.
	GetActiveUser (see page 671)	Returns all of the user properties with or without preference.
	GetADMapping (see page 672)	Loads the CMS and Active Directory name.
	GetAllCmsUsers (see page 672)	Loads all the CMS users.
	GetAllCustomProperties (see page 673)	Returns array of all the custom properties defined in the system.
	GetAllUserByCustomProperty (see page 673)	Returns all the users based on the given properties.
	GetAllUserGroups (see page 674)	Loads all of the user groups from the system. This method is overloaded.
	GetAllUsers (see page 675)	This method is overloaded.
	GetGroupsUserIsIn (see page 676)	Loads all the groups by user.
	GetMembershipUser (see page 676)	Returns user data based on the user's login name (username).
	GetUser (see page 677)	Loads the user details for the given user.
	GetUserbyUsername (see page 677)	Loads the user details for the given user.
	GetUserCustomProperties (see page 678)	Returns an array of custom properties for a given user's ID.
	GetUserEmailInfo (see page 678)	Returns user data for a given user's ID.
	GetUserForFolderAdmin (see page 679)	
	GetUserGroup (see page 679)	Loads a given group's details.
	GetUserGroupForFolderAdmin (see page 679)	Returns all the groups data for the administrator.
	GetUserPermissions (see page 680)	Loads the user permissions as a array of UserPermissionData.
	GetUserPreference (see page 680)	Loads the preferences for the given user.
	GetUsers (see page 681)	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.
	GetUsersForTask (see page 682)	Loads the users that have permissions to the given content (see page 337).
	GetUsersNotInGroup (see page 683)	Loads all the users that are not in the given group.
	IsARoleMember (see page 683)	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.
	IsARoleMemberForContent (see page 686)	Determines whether a Role Member is allowed to work with a given piece of content (see page 337) as an administrator or not. Administrations privileges are for the given piece of content (see page 337) 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 (see page 686)	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.
	LogInUser (see page 687)	Used to log in the user.
	ResetMembershipUserPassword (see page 687)	Reset the password for the given membership user.
	UnsubscribeUser (see page 687)	Unsubscribes a user from the system.
	UpdateCustomPropertiesItemOrder (see page 688)	Updates the order of available custom properties.
	UpdateUser (see page 688)	
	UpDateUserGroup (see page 689)	Updates the user group details.

	UpdateUserPreferences (see page 690)	Updates the given user's preference data.
	UserObject (see page 690)	Returns user data based on the user's ID or by the user's login or username. This function is overloaded.

Topics

Name	Description
User Methods (see page 663)	The methods of the User class are listed here.

Legend

	Method

User Methods

	Name	Description
	ActivateUserAccount (see page 665)	Activates a deleted (deactivated) user into the current/active state using the UserId or the Username and GUID (AccountID) parameters. This method is overloaded.
	AddMembershipUser (see page 666)	Creates new membership user.
	AddNewUser (see page 666)	Creates new membership user.
	AddUser (see page 667)	Creates new cms user
	AddUserGroup (see page 667)	Creates new user group.
	AddUserToGroup (see page 668)	Adds an existing user to an existing group.
	AutoLogInUser (see page 668)	
	CreateBlankUserObject (see page 668)	This method is overloaded.
	DeleteUserByID (see page 669)	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 670)	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 670)	Removes a user from the designated group.
	DeleteUserGroup (see page 671)	Removes the group from the system permanently.
	EditUserCustomProperties (see page 671)	This method creates the editable fields(UI) for user's custom properties.
	GetActiveUser (see page 671)	Returns all of the user properties with or without preference.
	GetADMapping (see page 672)	Loads the CMS and Active Directory name.
	GetAllCmsUsers (see page 672)	Loads all the CMS users.
	GetAllCustomProperties (see page 673)	Returns array of all the custom properties defined in the system.
	GetAllUserByCustomProperty (see page 673)	Returns all the users based on the given properties.
	GetAllUserGroups (see page 674)	Loads all of the user groups from the system. This method is overloaded.
	GetAllUsers (see page 675)	This method is overloaded.
	GetGroupsUserIsIn (see page 676)	Loads all the groups by user.
	GetMembershipUser (see page 676)	Returns user data based on the user's login name (username).
	GetUser (see page 677)	Loads the user details for the given user.
	GetUserbyUsername (see page 677)	Loads the user details for the given user.
	GetUserCustomProperties (see page 678)	Returns an array of custom properties for a given user's ID.
	GetUserEmailInfo (see page 678)	Returns user data for a given user's ID.

	GetUserForFolderAdmin (see page 679)	
	GetUserGroup (see page 679)	Loads a given group's details.
	GetUserGroupForFolderAdmin (see page 679)	Returns all the groups data for the administrator.
	GetUserPermissions (see page 680)	Loads the user permissions as a array of UserPermissionData.
	GetUserPreference (see page 680)	Loads the preferences for the given user.
	GetUsers (see page 681)	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.
	GetUsersForTask (see page 682)	Loads the users that have permissions to the given content (see page 337).
	GetUsersNotInGroup (see page 683)	Loads all the users that are not in the given group.
	IsARoleMember (see page 683)	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.
	IsARoleMemberForContent (see page 686)	Determines whether a Role Member is allowed to work with a given piece of content (see page 337) as an administrator or not. Administrations privileges are for the given piece of content (see page 337) 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 (see page 686)	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.
	LogInUser (see page 687)	Used to log in the user.
	ResetMembershipUserPassword (see page 687)	Reset the password for the given membership user.
	UnsubscribeUser (see page 687)	Unsubscribes a user from the system.
	UpdateCustomPropertiesItemOrder (see page 688)	Updates the order of available custom properties.
	UpdateUser (see page 688)	
	UpDateUserGroup (see page 689)	Updates the user group details.
	UpdateUserPreferences (see page 690)	Updates the given user's preference data.
	UserObject (see page 690)	Returns user data based on the user's ID or by the user's login or username. This function is overloaded.

Legend

	Method

User Methods

The methods of the User class are listed here.

Public Methods

	Name	Description
	ActivateUserAccount (see page 665)	Activates a deleted (deactivated) user into the current/active state using the UserId or the Username and GUID (AccountID) parameters. This method is overloaded.
	AddMembershipUser (see page 666)	Creates new membership user.
	AddNewUser (see page 666)	Creates new membership user.
	AddUser (see page 667)	Creates new cms user
	AddUserGroup (see page 667)	Creates new user group.
	AddUserToGroup (see page 668)	Adds an existing user to an existing group.
	AutoLogInUser (see page 668)	

	CreateBlankUserObject (see page 668)	This method is overloaded.
	DeleteUserByID (see page 669)	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 670)	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 670)	Removes a user from the designated group.
	DeleteUserGroup (see page 671)	Removes the group from the system permanently.
	EditUserCustomProperties (see page 671)	This method creates the editable fields(UI) for user's custom properties.
	GetActiveUser (see page 671)	Returns all of the user properties with or without preference.
	GetADMapping (see page 672)	Loads the CMS and Active Directory name.
	GetAllCmsUsers (see page 672)	Loads all the CMS users.
	GetAllCustomProperties (see page 673)	Returns array of all the custom properties defined in the system.
	GetAllUserByCustomProperty (see page 673)	Returns all the users based on the given properties.
	GetAllUserGroups (see page 674)	Loads all of the user groups from the system. This method is overloaded.
	GetAllUsers (see page 675)	This method is overloaded.
	GetGroupsUserIsIn (see page 676)	Loads all the groups by user.
	GetMembershipUser (see page 676)	Returns user data based on the user's login name (username).
	GetUser (see page 677)	Loads the user details for the given user.
	GetUserbyUsername (see page 677)	Loads the user details for the given user.
	GetUserCustomProperties (see page 678)	Returns an array of custom properties for a given user's ID.
	GetUserEmailInfo (see page 678)	Returns user data for a given user's ID.
	GetUserForFolderAdmin (see page 679)	
	GetUserGroup (see page 679)	Loads a given group's details.
	GetUserGroupForFolderAdmin (see page 679)	Returns all the groups data for the administrator.
	GetUserPermissions (see page 680)	Loads the user permissions as a array of UserPermissionData.
	GetUserPreference (see page 680)	Loads the preferences for the given user.
	GetUsers (see page 681)	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.
	GetUsersForTask (see page 682)	Loads the users that have permissions to the given content (see page 337).
	GetUsersNotInGroup (see page 683)	Loads all the users that are not in the given group.
	IsARoleMember (see page 683)	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.
	IsARoleMemberForContent (see page 686)	Determines whether a Role Member is allowed to work with a given piece of content (see page 337) as an administrator or not. Administrations privileges are for the given piece of content (see page 337) 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 (see page 686)	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.
	LogInUser (see page 687)	Used to log in the user.
	ResetMembershipUserPassword (see page 687)	Reset the password for the given membership user.

	UnsubscribeUser (see page 687)	Unsubscribes a user from the system.
	UpdateCustomPropertiesItemOrder (see page 688)	Updates the order of available custom properties.
	UpdateUser (see page 688)	
	UpDateUserGroup (see page 689)	Updates the user group details.
	UpdateUserPreferences (see page 690)	Updates the given user's preference data.
	UserObject (see page 690)	Returns user data based on the user's ID or by the user's login or username. This function is overloaded.

Legend

	Method

ActivateUserAccount Method

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 (int) (see page 665)	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 666)	Activates the deleted (deactivated) user into the current/active state using the Username and GUID (AccountID) parameter. This function is overloaded. See Also: ActivateUserAccount (Integer)

User.ActivateUserAccount Method (int)

Activates a deleted (deactivated) user into the current/active state using the UserId.

This function is overloaded. See Also: ActivateUserAccount (String String)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "ActivateUserAccount By UserID", Description = "")]
public bool ActivateUserAccount(int UserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "ActivateUserAccount By UserID", Description = "")>
Public Function ActivateUserAccount(UserId As Integer) As bool
```

Parameters

Parameters	Description
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 (Integer)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "ActivateUserAccount By UserName and AccountID", Description = "")]
public bool ActivateUserAccount(string Username, string AccountID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "ActivateUserAccount By UserName and AccountID", Description = "")>
Public Function ActivateUserAccount(Username As string, AccountID As string) As bool
```

Parameters

Parameters	Description
Username	The login name for the user
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddMembershipUser", Description = "")]
public UserData AddMembershipUser(UserData UserData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddMembershipUser", Description = "")>
Public Function AddMembershipUser(UserData As UserData) As UserData
```

Parameters

Parameters	Description
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

Creates new membership user.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddNewUser", Description = "")]
```

```
[Obsolete("Use \"AddMembershipUser\" instead.")]
public UserData AddNewUser(UserData UserData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddNewUser", Description = "")>
<Obsolete("Use \"AddMembershipUser\" instead.")>
Public Function AddNewUser(UserData As UserData) As UserData
```

Parameters

Parameters	Description
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.AddUser Method

Creates new cms user

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddUser", Description = "Creates new cms user.")]
public UserData AddUser(UserData user);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddUser", Description = "Creates new cms user.")>
Public Function AddUser(user As UserData) As UserData
```

Parameters

Parameters	Description
user	Collection (see page 63) of User'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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddUserGroup", Description = "")]
public void AddUserGroup(string GroupName, string GroupPath, string GroupDomain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddUserGroup", Description = "")>
Public Function AddUserGroup(GroupName As string, GroupPath As string, GroupDomain As string)
As void
```

Parameters

Parameters	Description
GroupName	The name of the group.
GroupPath	Group path, usually required for Active Directory, otherwise CMS group assumed.
GroupDomain	Group domain, usually required for Active Directory.

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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AddUserToGroup", Description = "")]
public void AddUserToGroup(int UserId, int GroupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AddUserToGroup", Description = "")>
Public Function AddUserToGroup(UserId As Integer, GroupId As Integer) As void
```

Parameters

Parameters	Description
UserId	The user's ID.
GroupId	The required group's ID.

Remarks

A user with administrator privileges must be logged in to perform this operation.

User.AutoLogInUser Method

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "AutoLogInUser", Description = "")]
public UserData AutoLogInUser(string SAMusername, string domain, string ServerName);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "AutoLogInUser", Description = "")>
Public Function AutoLogInUser(SAMusername As string, domain As string, ServerName As string)
As UserData
```

Parameters

Parameters	Description
SAMusername	
domain	
ServerName	

CreateBlankUserObject Method

This method is overloaded.

Overload List

	Name	Description
	User.CreateBlankUserObject () (see page 669)	Returns the instance of the user data. This function is overloaded. See Also: CreateBlankUserObject (UserData)
	User.CreateBlankUserObject (ref UserData) (see page 669)	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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "CreateBlankUserObject", Description = "")]
public UserData CreateBlankUserObject();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "CreateBlankUserObject", Description = "")>
Public Function CreateBlankUserObject() As UserData
```

Returns

Ektron.Cms.UserData

User.CreateBlankUserObject Method (ref UserData)

Sets the given user data values into the UserData class.

This function is overloaded. See Also: CreateBlankUserObject

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "CreateBlankUserObject By UserData", Description = "")]
public void CreateBlankUserObject(ref UserData UserData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "CreateBlankUserObject By UserData", Description = "")>
Public Function CreateBlankUserObject(UserData As ref UserData) As void
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteUserByID", Description = "")]
public void DeleteUserByID(int UserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteUserByID", Description = "")>
Public Function DeleteUserByID(UserId As Integer) As void
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteUserByIds", Description = "")]
public void DeleteUserByIds(string Ids);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteUserByIds", Description = "")>
Public Function DeleteUserByIds(Ids As string) As void
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteUserFromGroup", Description = "")]
public void DeleteUserFromGroup(int UserId, int GroupId, bool ReturnDeleted);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteUserFromGroup", Description = "")>
Public Function DeleteUserFromGroup(UserId As Integer, GroupId As Integer, ReturnDeleted As
bool) As void
```

Parameters

Parameters	Description
UserId	The user's ID.
GroupId	The group's ID.
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "DeleteUserGroup", Description = "")]
public void DeleteUserGroup(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "DeleteUserGroup", Description = "")>
Public Function DeleteUserGroup(Id As Integer) As void
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "EditUserCustomProperties", Description = "")]
public string EditUserCustomProperties(int UserId, bool ReadOnly);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "EditUserCustomProperties", Description = "")>
Public Function EditUserCustomProperties(UserId As Integer, ReadOnly As bool) As string
```

Parameters

Parameters	Description
UserId	User's ID. If the user ID is not specified then it will return empty fields.
ReadOnly	

Returns

String

User.GetActiveUser Method

Returns all of the user properties with or without preference.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetActiveUser", Description = "")]
public UserData GetActiveUser(int Id, bool Preference);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetActiveUser", Description = "")>
Public Function GetActiveUser(Id As Integer, Preference As bool) As UserData
```

Parameters

Parameters	Description
Id	The user's ID.
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 name.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetADMapping", Description = "")]
public AdMappingData[] GetADMapping(int CallerId, string MapType, int Map, int Exclude, int
Editable);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetADMapping", Description = "")>
Public Function GetADMapping(CallerId As Integer, MapType As string, Map As Integer, Exclude
As Integer, Editable As Integer) As AdMappingData[]
```

Parameters

Parameters	Description
CallerId	Who is calling this function.
MapType	Possible value is “userprop” (user properties).
Map	Ektron.Cms.Common.EkEnumeration.MapTypes (include, exclude and dontconsider are the possible values).
Exclude	Flag (0 – false, 1-true).
Editable	Flag(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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllCmsUsers", Description = "")]
public UserData[] GetAllCmsUsers(string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllCmsUsers", Description = "")>
Public Function GetAllCmsUsers(OrderBy As string) As UserData[]
```

Parameters

Parameters	Description
`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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllCustomProperties", Description = "")]
public CustomAttribute[] GetAllCustomProperties();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllUserByCustomProperty", Description = "")]
public UserData[]
GetAllUserByCustomProperty(Ektron.Cms.Common.EkEnumeration.ObjectPropertyValueTypes PropType,
string PropValues);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllUserByCustomProperty", Description = "")>
Public Function GetAllUserByCustomProperty(PropType As
Ektron.Cms.Common.EkEnumeration.ObjectPropertyValueTypes, PropValues As string) As UserData[]
```

Parameters

Parameters	Description
PropType	
PropValues	Required property

Remarks

A user with administrator privileges must be logged in to perform this operation.

GetAllUserGroups Method

Loads all of the user groups from the system. This method is overloaded.

Overload List

	Name	Description
	User.GetAllUserGroups (EkEnumeration.UserTypes, string) (see page 674)	Loads all of the user groups from the system. This function is overloaded. See Also: GetAllUserGroups (String)
	User.GetAllUserGroups (string) (see page 674)	Loads all of the user groups from the system. This function is overloaded. See Also: GetAllUserGroups (EkEnumeration.UserTypes String)

User.GetAllUserGroups Method (EkEnumeration.UserTypes, string)

Loads all of the user groups from the system.

This function is overloaded. See Also: GetAllUserGroups (String)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllUserGroups By GroupType and OrderBy", Description = "")]
public UserGroupData[] GetAllUserGroups(EkEnumeration.UserTypes GroupType, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllUserGroups By GroupType and OrderBy", Description = "")>
Public Function GetAllUserGroups(GroupType As EkEnumeration.UserTypes, OrderBy As string) As
UserGroupData[]
```

Parameters

Parameters	Description
GroupType	GroupType: 1 = Membership (see page 212); 0 = CMSUser
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)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllUserGroups", Description = "")]
public UserGroupData[] GetAllUserGroups(string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllUserGroups", Description = "")>
Public Function GetAllUserGroups(OrderBy As string) As UserGroupData[]
```

Parameters

Parameters	Description
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 method is overloaded.

Overload List

	Name	Description
	User.GetAllUsers (int, int) (see page 675)	Gets the User Group Data of all the allowed users in the folder specified.
	User.GetAllUsers (ref UserRequestData) (see page 676)	Returns the list of users from the database based on the UserRequestData class.

User.GetAllUsers Method (int, int)

Gets the User Group Data of all the allowed users in the folder specified.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllUsers By CallerId and FolderID", Description = "")]
public UserGroupData[] GetAllUsers(int CallerID, int FolderId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllUsers By CallerId and FolderID", Description = "")>
Public Function GetAllUsers(CallerID As Integer, FolderId As Integer) As UserGroupData[]
```

Parameters

Parameters	Description
CallerID	The user ID that is logged on to the system.
FolderId	The ID of the folder.

Returns

Ektron.Cms.UserGroupData()

Remarks

User login with administrator privileges are required to perform this operation.

User.GetAllUsers Method (ref UserRequestData)

Returns the list of users from the database based on the UserRequestData class.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetAllUsers by UserRequest", Description = "")]
public UserData[] GetAllUsers(ref UserRequestData userRequest);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetAllUsers by UserRequest", Description = "")>
Public Function GetAllUsers(userRequest As ref UserRequestData) As UserData[]
```

Parameters

Parameters	Description
userRequest	list of constraints to load users

Returns

An array of users as UserData.

User.GetGroupsUserIsIn Method

Loads all the groups by user.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetGroupsUserIsIn", Description = "")]
public GroupData[] GetGroupsUserIsIn(int UserId, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetGroupsUserIsIn", Description = "")>
Public Function GetGroupsUserIsIn(UserId As Integer, OrderBy As string) As GroupData[]
```

Parameters

Parameters	Description
UserId	The user's ID.
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).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetMembershipUser", Description = "")]
public UserData GetMembershipUser(string Username);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetMembershipUser", Description = "")>
Public Function GetMembershipUser(Username As string) As UserData
```

Parameters

Parameters	Description
Username	User's login name (username). This is required

Returns

Ektron.Cms.UserData

Remarks

User login with administrator privileges are required to perform this operation.

User.GetUser Method

Loads the user details for the given user.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUser", Description = "")]
public UserData GetUser(int Id, bool Preference, bool ReturnDeleted);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUser", Description = "")>
Public Function GetUser(Id As Integer, Preference As bool, ReturnDeleted As bool) As UserData
```

Parameters

Parameters	Description
Id	The user ID.
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUserbyUsername", Description = "")]
public UserData GetUserbyUsername(string username);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUserbyUsername", Description = "")>
Public Function GetUserbyUsername(username As string) As UserData
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUserCustomProperties", Description = "")]
public CustomAttribute[] GetUserCustomProperties(int UserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUserCustomProperties", Description = "")>
Public Function GetUserCustomProperties(UserId As Integer) As CustomAttribute[]
```

Parameters

Parameters	Description
UserId	User'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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUserEmailInfo", Description = "")]
public UserData GetUserEmailInfo(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUserEmailInfo", Description = "")>
Public Function GetUserEmailInfo(Id As Integer) As UserData
```

Parameters

Parameters	Description
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")]
[WebMethod(MessageName = "GetUserForFolderAdmin", Description = "")]
public UserData GetUserForFolderAdmin(int FolderId, int Id, bool Preference, bool
ReturnDeleted);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUserForFolderAdmin", Description = "")>
Public Function GetUserForFolderAdmin(FolderId As Integer, Id As Integer, Preference As bool,
ReturnDeleted As bool) As UserData
```

Parameters

Parameters	Description
FolderId	
Id	
Preference	
ReturnDeleted	

User.GetUserGroup Method

Loads a given group's details.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUserGroup", Description = "")]
public UserGroupData GetUserGroup(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUserGroup", Description = "")>
Public Function GetUserGroup(Id As Integer) As UserGroupData
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUserGroupForFolderAdmin", Description = "")]
public UserGroupData GetUserGroupForFolderAdmin(int FolderId, int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUserGroupForFolderAdmin", Description = "")>
Public Function GetUserGroupForFolderAdmin(FolderId As Integer, Id As Integer) As UserGroupData
```

Parameters

Parameters	Description
FolderId	Folder ID
Id	User ID

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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUserPermissions", Description = "")]
public UserPermissionData[] GetUserPermissions(int Id, string ItemType, int UserId, string
UserGroupList, EkContent.PermissionUserType PermissionType, EkContent.PermissionRequestType
PermissionRequest);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUserPermissions", Description = "")>
Public Function GetUserPermissions(Id As Integer, ItemType As string, UserId As Integer,
UserGroupList As string, PermissionType As EkContent.PermissionUserType, PermissionRequest As
EkContent.PermissionRequestType) As UserPermissionData[]
```

Parameters

Parameters	Description
Id	the item's ID
ItemType	The item's type
UserId	The user's ID
UserGroupList	the group list
PermissionType	PermissionUserType
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUserPreference", Description = "")]
public UserPreferenceData GetUserPreference(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUserPreference", Description = "")>
Public Function GetUserPreference(Id As Integer) As UserPreferenceData
```

Parameters

Parameters	Description
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

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 (int, string) (see page 681)	Loads the all the users from the given group.
	User.GetUsers (string, string) (see page 682)	Returns an array of user group data based on the userName and userNameType.

User.GetUsers Method (int, string)

Loads the all the users from the given group.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUsers By GroupID and OrderBy", Description = "")]
public UserData[] GetUsers(int GroupId, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUsers By GroupID and OrderBy", Description = "")>
Public Function GetUsers(GroupId As Integer, OrderBy As string) As UserData[]
```

Parameters

Parameters	Description
GroupId	The group's ID
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUsers", Description = "")]
public UserGroupData[] GetUsers(string userNames, string userNameTypes);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUsers", Description = "")>
Public Function GetUsers(userNames As string, userNameTypes As string) As UserGroupData[]
```

Parameters

Parameters	Description
userNames	The user's name. (Required)
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 (see page 337).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUsersForTask", Description = "")]
public UserGroupData[] GetUsersForTask(int CallerID, int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUsersForTask", Description = "")>
Public Function GetUsersForTask(CallerID As Integer, ContentID As Integer) As UserGroupData[]
```

Parameters

Parameters	Description
CallerID	The user ID that is logged on to the system.
ContentID	The ID of the content (see page 337). (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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "GetUsersNotInGroup", Description = "")]
public UserData[] GetUsersNotInGroup(int Id, string OrderBy, int GroupType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "GetUsersNotInGroup", Description = "")>
Public Function GetUsersNotInGroup(Id As Integer, OrderBy As string, GroupType As Integer) As
UserData[]
```

Parameters

Parameters	Description
Id	The group ID.
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.

IsARoleMember Method

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 684)	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) and CommonApi.IsARoleMember Method (Integer).

	User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, int, bool) (see page 684)	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) and CommonApi.IsARoleMember Method (Integer).
	User.IsARoleMember (int) (see page 685)	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) and CommonApi.IsARoleMember Method (Integer, Integer, Boolean).
	User.IsARoleMember (int, int, bool) (see page 685)	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) and CommonApi.IsARoleMember Method (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: CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean), CommonApi.IsARoleMember Method (Integer, Integer, Boolean) and CommonApi.IsARoleMember Method (Integer).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "IsARoleMember by CMSroleID", Description = "")]
public bool IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "IsARoleMember by CMSroleID", Description = "")>
Public Function IsARoleMember(RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds) As bool
```

Parameters

Parameters	Description
RoleId	

User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, int, 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) and CommonApi.IsARoleMember Method (Integer).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "IsARoleMember by CmsRoleID, UserID and GroupFlag", Description = "")]
public bool IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId, int UserId, bool
GroupFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "IsARoleMember by CmsRoleID, UserID and GroupFlag", Description = "")>
Public Function IsARoleMember(RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds, UserId As
Integer, GroupFlag As bool) As bool
```

Parameters

Parameters	Description
RoleId	
GroupFlag	
nUserId	

User.IsARoleMember Method (int)

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) and CommonApi.IsARoleMember Method (Integer, Integer, Boolean).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "IsARoleMember by RoleID", Description = "")]
public bool IsARoleMember(int RoleId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "IsARoleMember by RoleID", Description = "")>
Public Function IsARoleMember(RoleId As Integer) As bool
```

Parameters

Parameters	Description
RoleId	

User.IsARoleMember Method (int, int, 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) and CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds).

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "IsARoleMember by RoleID, UserId and GroupFlag", Description = "")]
public bool IsARoleMember(int RoleId, int UserId, bool GroupFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "IsARoleMember by RoleID, UserId and GroupFlag", Description = "")>
Public Function IsARoleMember(RoleId As Integer, UserId As Integer, GroupFlag As bool) As bool
```

Parameters

Parameters	Description
RoleId	
GroupFlag	
nUserId	

User.IsARoleMemberForContent Method

Determines whether a Role Member is allowed to work with a given piece of content (see page 337) as an administrator or not. Administrations privileges are for the given piece of content (see page 337) 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#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "IsARoleMemberForContent", Description = "")]
public bool IsARoleMemberForContent(int RoleId, int ContentId, int UserId, bool GroupFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "IsARoleMemberForContent", Description = "")>
Public Function IsARoleMemberForContent(RoleId As Integer, ContentId As Integer, UserId As
Integer, GroupFlag As bool) As bool
```

Parameters

Parameters	Description
RoleId	
ContentId	The numeric content (see page 337) ID for which you are giving the Role Member privileges.
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "IsARoleMemberForFolder", Description = "")]
public bool IsARoleMemberForFolder(int RoleId, int FolderId, int UserId, bool GroupFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "IsARoleMemberForFolder", Description = "")>
Public Function IsARoleMemberForFolder(RoleId As Integer, FolderId As Integer, UserId As
Integer, GroupFlag As bool) As bool
```

Parameters

Parameters	Description
RoleId	
FolderId	The numeric folder ID for which you are giving the Role Member privileges.
UserId	

GroupFlag	

User.LogInUser Method

Used to log in the user.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "LogInUser", Description = "")]
public UserData LogInUser(string username, string Password, string ServerName, string Domain,
string Protocol);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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
username	The system username.
Password	The system password.
ServerName	The system server name.
Domain	The user's domain, required only when users login using Active Directory.
Protocol	Required only when using a secure login or when you are using any valid Active Directory protocols.

Returns

Ektron.Cms.UserData

User.ResetMembershipUserPassword Method

Reset the password for the given membership user.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "ResetMembershipUserPassword", Description = "")]
public string ResetMembershipUserPassword(string Username);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "ResetMembershipUserPassword", Description = "")>
Public Function ResetMembershipUserPassword(Username As string) As string
```

Parameters

Parameters	Description
Username	User's username to be reset

Returns

String

Remarks

User login is required to perform this operation.

User.UnsubscribeUser Method

Unsubscribes a user from the system.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UnsubscribeUser", Description = "")]
public bool UnsubscribeUser(string Username, string Password);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UnsubscribeUser", Description = "")>
Public Function UnsubscribeUser(Username As string, Password As string) As bool
```

Parameters

Parameters	Description
Username	The user's username
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateCustomPropertiesItemOrder", Description = "")]
public bool UpdateCustomPropertiesItemOrder(string ItemOrder);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UpdateCustomPropertiesItemOrder", Description = "")>
Public Function UpdateCustomPropertiesItemOrder(ItemOrder As string) As bool
```

Parameters

Parameters	Description
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.

UpdateUser Method

Overload List

	Name	Description
	User.UpdateUser (Collection) (see page 689)	Updates the user details.
	User.UpdateUser (UserData) (see page 689)	Updates the user details.

User.UpdateUser Method (Collection)

Updates the user details.

C#

```
public void UpdateUser(Collection data);
```

Visual Basic

```
Public Function UpdateUser(data As Collection) As void
```

Parameters

Parameters	Description
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.

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UpdateUser", Description = "")]
public void UpdateUser(UserData data);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UpdateUser", Description = "")>
Public Function UpdateUser(data As UserData) As void
```

Parameters

Parameters	Description
data	A class of type UserData.

Remarks

User login with administrator privileges are required to perform this operation.

User.UpDateUserGroup Method

Updates the user group details.

C#

```
public void UpDateUserGroup(UserData data);
```

Visual Basic

```
Public Function UpDateUserGroup(data As UserData) As void
```

Parameters

Parameters	Description
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.

C#

```
public void UpdateUserPreferences(int Id, Collection data);
```

Visual Basic

```
Public Function UpdateUserPreferences(Id As Integer, data As Collection) As void
```

Parameters

Parameters	Description
Id	The user's ID.
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

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 (int) (see page 690)	Returns user data based on the user's ID. This function is overloaded. See Also: UserObject (String)
	User.UserObject (string) (see page 691)	Returns user data based on the user's login or username. This function is overloaded. See Also: UserObject (Integer)

User.UserObject Method (int)

Returns user data based on the user's ID.

This function is overloaded. See Also: UserObject (String)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UserObject By User ID", Description = "")]
public UserData UserObject(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UserObject By User ID", Description = "")>
Public Function UserObject(Id As Integer) As UserData
```

Parameters

Parameters	Description
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 (Integer)

C#

```
[SoapHeader("AuthHeader")]
[WebMethod(MessageName = "UserObject By UserName", Description = "")]
public UserData UserObject(string UserName);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<WebMethod(MessageName = "UserObject By UserName", Description = "")>
Public Function UserObject(UserName As string) As UserData
```

Parameters

Parameters	Description
sUserName	User's login or username.

Returns

Ektron.Cms.UserData

Symbol Reference

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	CommentData (see page 692)	This is class CommentData.

Legend

	Class

CommentData Class

Class Hierarchy

CommentData

C#

```
public class CommentData;
```

Visual Basic

```
Public Class CommentData
```

File

CommentData.vb

Description

This is class CommentData.

Index

.

.NET Assemblies 329

A

ActiveDirectoryUser class 545
- about ActiveDirectoryUser class 545
- ActiveDirectoryUser methods 547
- AddADGroupToCMS 548
- DeSynchUserGroups 548
- DeSynchUsers 548
- GetADMapping 549
- GetADStatus 549
- GetAvailableADGroups 550
- GetAvailableADUsers 550
- GetCMSGroupsToSync 551
- GetCMSRelationshipsToSync 551
- GetCMSUsersToSync 551
- GetDomains 552
- MapCMSUserGroupToAD 552
- MapCMSUserToAD 553
- UpdateADMapping 553
- UpdateADUsersGroups 553
- UpdateUsersInfoFromAD 554

ActiveTopics class 9
- about ActiveTopics class 9
- ActiveTopics methods 10
- ActiveTopics properties 11
- BoardID 12
- CacheInterval 12
- Fill 11
- ToString 11
- Wrap 11

ADUser class 403
- about ADUser class 403
- AddADGroupToCMS 405
- AddADUsersToCMSByUsername 406
- ADUser methods 404
- DeSynchUserGroups 406
- DeSynchUsers 406
- GetADMapping 407
- GetADStatus 407
- GetAvailableADGroups 407
- GetAvailableADUsers 408
- GetCMSGroupsToSync 408
- GetCMSRelationshipsToSync 409
- GetCMSUsersToSync 409
- GetDomains 409
- MapCMSUserGroupToAD 410
- MapCMSUserToAD 410
- SynchCMSGroupsToAD 410
- SynchCMSRelationshipsToAD 411
- SynchCMSUsersToAD 411
- UpdateADMapping 412
- UpdateADUsersGroups 412
- UpdateUsersInfoFromAD 412

Analytics class 330
- about Analytics class 330
- Analytics methods 332
- Analytics properties 334
- AnalyticsEnabled 334
- AppendToIISLog 335
- GetSessionRank 332
- HardDumpAnalyticsBuffer 332
- InMemoryContentHitsData 335
- QueryAnalytics 332
- RecordsBeforeWrite 335
- SoftDumpAnalyticsBuffer 333
- TimeBeforeWrite 335
- TrackVisit 333, 334
- TrackVisit method 333

AnalyticsTracker class 12
- about AnalyticsTracker class 12
- AnalyticsTracker methods 13
- AnalyticsTracker properties 14
- DefaultContentID 14

DynamicParameter 14
EnableAnalytics 15
Wrap 14
API 329
API namespace 329
Classes 443
API.Analytics 330
API.Analytics namespace 330
Classes 330
API.Analytics.Analytics 330
API.Analytics.Analytics.AnalyticsEnabled 334
API.Analytics.Analytics.AppendToIISLog 335
API.Analytics.Analytics.GetSessionRank 332
API.Analytics.Analytics.HardDumpAnalyticsBuffer 332
API.Analytics.Analytics.InMemoryContentHitsData 335
API.Analytics.Analytics.QueryAnalytics 332
API.Analytics.Analytics.RecordsBeforeWrite 335
API.Analytics.Analytics.SoftDumpAnalyticsBuffer 333
API.Analytics.Analytics.TimeBeforeWrite 335
API.Analytics.Analytics.TrackVisit 333, 334
API.BusinessRule 444
API.BusinessRule.GetBusinessRules 445
API.Calendar 336
API.Calendar namespace 336
Classes 336
API.Calendar.Calendar 336
API.Calendar.CalendarEvent 337
API.Calendar.CalendarEventType 337
API.Content 337
API.Content namespace 337
Classes 338
API.Content.Asset 338
API.Content.Asset.GetAssetMgtConfigInfo 340
API.Content.Asset.GetAssetMgtInfo 340
API.Content.Asset.GetAssetSupertypes 340
API.Content.Asset.GetCreateDragDropSnippet 341
API.Content.Asset.GetCreateEktronExplorerSnippet 341
API.Content.Asset.GetCreateSnippet 341
API.Content.Asset.GetDragDropWOSnippet 341
API.Content.Asset.GetEditSnippet 342
API.Content.Asset.GetMyAssetMgtID 342
API.Content.Asset.GetViewUrl 342
API.Content.Asset.MakeCopyDMSAsset 343
API.Content.Asset.SetAssetMgtConfigInfo 343
API.Content.Blog 344
API.Content.Blog.AddBlog 346
API.Content.Blog.AddCommentForPost 346
API.Content.Blog.AddPost 346
API.Content.Blog.DeleteBlog 347
API.Content.Blog.DeletePost 347
API.Content.Blog.GetBlankBlogPostData 347
API.Content.Blog.GetBlog 348, 349
API.Content.Blog.GetBlogData 350, 351
API.Content.Blog.GetBlogDataForPost 352
API.Content.Blog.GetBlogPostData 352
API.Content.Blog.GetBlogRoll 353
API.Content.Blog.GetPost 354
API.Content.Content 354
API.Content.Content.AddContent 360, 361
API.Content.Content.AddTaxonomyItem 362
API.Content.Content.AddTaxonomySyncFolder 363
API.Content.Content.AddXmlConfiguration 363
API.Content.Content.CheckOutContent 363
API.Content.Content.CopyContentToFolder 364
API.Content.Content.CreateTaxonomy 364
API.Content.Content.DeleteContentItem 364
API.Content.Content.DeleteTaxonomy 365
API.Content.Content.DeleteXmlConfiguration 365
API.Content.Content.GetAddViewLanguage 365
API.Content.Content.GetAllComments 365
API.Content.Content.GetAllTemplates 366
API.Content.Content.GetAllUnassignedItemApprovals 366
API.Content.Content.GetAllXmlConfigurations 367
API.Content.Content.GetChildContent 367
API.Content.Content.GetContent 368
API.Content.Content.GetContentByHistory 369
API.Content.Content.GetContentDisplayLink 369
API.Content.Content.GetContentForEditing 370

API.Content.Content.GetContentReport 370
API.Content.Content.GetContentState 370
API.Content.Content.GetContentStatus 371
API.Content.Content.GetDomain 371
API.Content.Content.GetExpireContent 371
API.Content.Content.GetTaxonomyIdByPath 372
API.Content.Content.LoadTaxonomy 372
API.Content.Content.MoveContentToFolder 372
API.Content.Content.PublishContent 373
API.Content.Content.ReadAllAssignedCategory 373
API.Content.Content.ReadCategoryByTaxonomy 374
API.Content.Content.ReadTaxonomy 374
API.Content.Content.RelicateXmlConfiguration 374
API.Content.Content.RemoveTaxonomyFolder 375
API.Content.Content.RemoveTaxonomyItem 375
API.Content.Content.RenameContent 375
API.Content.Content.ReOrderTaxonomyItems 375
API.Content.Content.RestoreHistoryContent 376
API.Content.Content.SaveContent 376
API.Content.Content.SubmitForDelete 377
API.Content.Content.UndoCheckout 377
API.Content.Content.UpdateContentMetaData 377
API.Content.Content.UpdateTaxonomy 378
API.Content.Content.UpdatexmlCollectionPackage 378
API.Content.Content.UpdateXmlConfiguration 378
API.Content.ContentRating 379
API.Content.ContentRating.GetContentRating 380
API.Content.ContentRating.GetContentRatingAddRating 381
API.Content.ContentRating.GetContentRatingResults 381
API.Content.ContentRating.GetContentRatingStatistics 382
API.Content.ContentRating.PurgeContentRatings 383
API.Content.Form 383
API.Content.Form.AssignFormTask 385
API.Content.Form.EmailFormFieldData 385
API.Content.Form.GetAllForms 386
API.Content.Form.GetForm 387
API.Content.Form.GetFormDataHistogram 387
API.Content.Form.GetFormFieldData 387, 388
API.Content.Form.GetFormFieldList 388
API.Content.Form.GetFormFieldQuestions 389
API.Content.Form.GetFormTitle 389
API.Content.Form.SubmitFormFieldData 389
API.Content.Form.UpdateFormFieldData 390
API.Content.ThreadedDiscussion 390
API.Content.ThreadedDiscussion.AddBoard 393
API.Content.ThreadedDiscussion.AddForum 393
API.Content.ThreadedDiscussion.AddTopic 394
API.Content.ThreadedDiscussion.DeleteBoard 395
API.Content.ThreadedDiscussion.DeleteForum 395
API.Content.ThreadedDiscussion.DeleteTopic 395
API.Content.ThreadedDiscussion.GetBoardCategories 396
API.Content.ThreadedDiscussion.GetBoardForums 396
API.Content.ThreadedDiscussion.GetDiscussionBoard 397
API.Content.ThreadedDiscussion.GetDiscussionBoardString 397
API.Content.ThreadedDiscussion.GetForum 398
API.Content.ThreadedDiscussion.GetRepliesForTopics 398
API.Content.ThreadedDiscussion.GetTopic 398
API.Content.ThreadedDiscussion.SearchReplies 399
API.CustomFields 446
API.CustomFields.AddNewEditableCustomFieldAssignments 450
API.CustomFields.BoolToYesNo 450
API.CustomFields.GetEditableCustomFieldAssignments 450
API.CustomFields.GetEnhancedMetadataArea 451
API.CustomFields.GetEnhancedMetadataScript 451
API.CustomFields.GetFieldsByFolder 451
API.CustomFields.GetSearchProperties 452, 453
API.CustomFields.Populate_AssetRequestObjectFromForm 453
API.CustomFields.ProcessCustomFields 454
API.CustomFields.SearchAllAssets 454
API.CustomFields.WriteFilteredMetadataForEdit 454
API.CustomFields.WriteFilteredMetadataForView 455
API.CustomFields.WriteLibrarySearchExtended 455
API.CustomFields.WriteMetadataDefaultForEdit 455
API.CustomFields.WriteMetadataTypeForView 456
API.CustomFields.WritePublicSearch 456
API.CustomFields.WritePublicSearchEx 457
API.DateTimeSelector 458

API.DateTimeSelector.ClientDateSeparator 462
API.DateTimeSelector.ClientLongDatePattern 462
API.DateTimeSelector.ClientLongTimePattern 463
API.DateTimeSelector.ClientShortDatePattern 463
API.DateTimeSelector.ClientShortTimePattern 463
API.DateTimeSelector.ClientTimeSeparator 463
API.DateTimeSelector.DisplayCultureDate 460
API.DateTimeSelector.DisplayCultureDateTime 460
API.DateTimeSelector.DisplayCultureTime 461
API.DateTimeSelector.ExtendedMeta 464
API.DateTimeSelector.FormElement 464
API.DateTimeSelector.FormName 464
API.DateTimeSelector.RequestInfo 464
API.DateTimeSelector.RoundMinutes 461
API.DateTimeSelector.ServerDateSeparator 465
API.DateTimeSelector.ServerShortDatePattern 465
API.DateTimeSelector.ServerShortTimePattern 465
API.DateTimeSelector.ServerTimeSeparator 465
API.DateTimeSelector.SpanAttributes 465
API.DateTimeSelector.SpanId 466
API.DateTimeSelector.TargetDate 466
API.DateTimeSelector.ValidationType 466
API.Folder 466
API.Folder.AddFolder 468
API.Folder.CreateFolder 468
API.Folder.DeleteFolderById 469
API.Folder.GetChildFolders 469
API.Folder.GetFolder 469
API.Folder.GetFolderDataWithPermission 470
API.Folder.GetFolderId 470, 471
API.Folder.GetFolderInfo 471
API.Folder.GetPath 471
API.Folder.RenameFolder 472
API.Folder.UpdateFolder 472
API.Font 472
API.Font.AddFont 473
API.Font.DeleteFont 474
API.Font.GetAllFonts 474
API.Font.GetFont 474
API.Font.GetFontConfigList 474
API.Font.UpdateFont 475
API.Library 475
API.Library.AddLibraryItem 477
API.Library.AddLoadBalanceItem 478
API.Library.DeleteLibraryItem 478
API.Library.DeleteLibraryItemById 478
API.Library.DeleteLoadBalanceItem 479
API.Library.GetAllChildLibItems 479
API.Library.GetAllLoadBalancePaths 480
API.Library.GetLibraryItem 481
API.Library.GetLibrarySettings 481
API.Library.GetLibraryTypes 482
API.Library.GetLoadBalancePath 482
API.Library.InternalLibrarySearch 482
API.Library.UpdateLibrary 483
API.Library.UpdateLibrarySettings 483
API.Library.UpdateLoadBalanceSettings 483
API.Library.UpdateQlinkTemplates 484
API.Localization 484
API.Localization.GetAddViewLanguage 487
API.Localization.GetAddViewLanguageForAllContent 487
API.Localization.GetAddViewLanguageForMenus 488
API.Localization.GetContentItems 488
API.Localization.GetFile 488
API.Localization.GetFiles 489
API.Localization.GetFilesInZipFile 489
API.Localization.GetFlagUrl 490
API.Localization.GetInterchangeFile 491
API.Localization.GetInterchangeFiles 491
API.Localization.GetInterchangeFilesInZipFile 492
API.Localization.GetJobs 492, 493
API.Localization.GetLocalizationUrl 493
API.Localization.GetTranslationUploadDirectory 493
API.Localization.StartExportContentForTranslation 493
API.Localization.StartExportFolderForTranslation 494
API.Localization.StartExportMenusForTranslation 494
API.Localization.StartImportTranslation 495
API.Metadata 495

API.Metadata.AddMetaDataType 497
API.Metadata.DeleteMetadataType 497
API.Metadata.GetContentMetadataList 498
API.Metadata.GetMetadataType 499
API.Metadata.GetMetaDataTypes 499
API.Metadata.UpdateMetadataType 499
API.Permissions 500
API.Permissions.AddItemPermission 501
API.Permissions.DeleteItemPermission 502
API.Permissions.DisableContentInheritance 502
API.Permissions.DisableFolderInheritance 502
API.Permissions.EnableContentInheritance 503
API.Permissions.EnableFolderInheritance 503
API.Permissions.GetUserPermissions 503, 504
API.Permissions.LoadPermissions 504
API.Permissions.UpdateItemPermission 505
API.Search 399
API.Search namespace 399
Classes 399
API.Search.SearchManager 400
API.Search.SearchManager.Execute 401, 402
API.Site 505
API.Site.BreakTaskPermissionInherit 507
API.Site.DeleteCookie 508
API.Site.GetAllActiveLanguages 508
API.Site.GetAllLanguages 508
API.Site.GetClientDayNames 508
API.Site.GetClientMonthNames 509
API.Site.GetCookieValue 509
API.Site.GetLanguage 509
API.Site.GetLanguages 510
API.Site.GetSiteVariables 510
API.Site.GetTaskPermission 510
API.Site.IsTaskPermissionInherited 511
API.Site.SetCookieValue 511
API.Site.SetTaskPermission 511
API.Site.UpdateSiteEnabledLanguages 512
API.Site.UpdateSiteVariables 512
API.SiteMap 512
API.SiteMap.GetFolderBreadcrumbPath 513
API.SiteMap.GetWebsiteMap 514
API.SiteMap.RenderFolderBreadcrumbHtml 514
API.SiteMap.RenderWebsiteMap 514
API.Subscription 515
API.Subscription.AddEmailFrom 518
API.Subscription.AddSubscription 518
API.Subscription.AddSubscriptionMessage 518
API.Subscription.DeleteEmailFrom 519
API.Subscription.DeleteSubscription 519
API.Subscription.DeleteSubscriptionMessage 519
API.Subscription.GetAllActiveSubscriptions 519
API.Subscription.GetAllEmailFrom 520
API.Subscription.GetAllSubscriptionMessages 520
API.Subscription.GetAllSubscriptions 520
API.Subscription.GetFolderInheritedFrom 521
API.Subscription.GetSubscription 521
API.Subscription.GetSubscriptionMessage 521
API.Subscription.GetSubscriptionMessagesForType 521
API.Subscription.GetSubscriptionPropertiesForContent 522
API.Subscription.GetSubscriptionPropertiesForFolder 522
API.Subscription.GetSubscriptionsForContent 522
API.Subscription.GetSubscriptionsForFolder 523
API.Subscription.UpdateEmailFrom 523
API.Subscription.UpdateSubscription 523
API.Subscription.UpdateSubscriptionMessage 524
API.User 402
API.User namespace 402
Classes 402
API.User.ADUser 403
API.User.ADUser.AddADGroupToCMS 405
API.User.ADUser.AddADUsersToCMSByUsername 406
API.User.ADUser.DeSynchUserGroups 406
API.User.ADUser.DeSynchUsers 406
API.User.ADUser.GetADMapping 407
API.User.ADUser.GetADStatus 407
API.User.ADUser.GetAvailableADGroups 407
API.User.ADUser.GetAvailableADUsers 408
API.User.ADUser.GetCMSGroupsToSync 408

API.User.ADUser.GetCMSRelationshipsToSync 409
API.User.ADUser.GetCMSUsersToSync 409
API.User.ADUser.GetDomains 409
API.User.ADUser.MapCMSUserGroupToAD 410
API.User.ADUser.MapCMSUserToAD 410
API.User.ADUser.SynchCMSGroupsToAD 410
API.User.ADUser.SynchCMSRelationshipsToAD 411
API.User.ADUser.SynchCMSUsersToAD 411
API.User.ADUser.UpdateADMapping 412
API.User.ADUser.UpdateADUsersGroups 412
API.User.ADUser.UpdateUsersInfoFromAD 412
API.User.User 413
API.User.User.ActivateUserAccount 419
API.User.User.AddMembershipUser 419
API.User.User.AddNewUser 420
API.User.User.AddRolePermission 420
API.User.User.AddUser 421
API.User.User.AddUserGroup 421
API.User.User.AddUserToGroup 421
API.User.User.AutoLogInUser 422
API.User.User.CreateBlankUserObject 422, 423
API.User.User.DeleteUserByID 423
API.User.User.DeleteUserByIds 423
API.User.User.DeleteUserFromGroup 423
API.User.User.DeleteUserGroup 424
API.User.User.EditUserCustomProperties 424
API.User.User.GetActiveUser 424
API.User.User.GetADMapping 424
API.User.User.GetAllCmsUsers 425
API.User.User.GetAllCustomProperties 425
API.User.User.GetAllUserByCustomProperty 426
API.User.User.GetAllUserGroups 426, 427
API.User.User.GetAllUsers 427, 428
API.User.User.GetGroupsUserIsIn 428
API.User.User.GetMembershipUser 428
API.User.User.GetUser 428
API.User.User.GetUserByUsername 429
API.User.User.GetUserCustomProperties 429
API.User.User.GetUserEmailInfo 429
API.User.User.GetUserForFolderAdmin 430
API.User.User.GetUserGroup 430
API.User.User.GetUserGroupByName 430
API.User.User.GetUserGroupForFolderAdmin 431
API.User.User.GetUserPermissions 431
API.User.User.GetUserPreference 431
API.User.User.GetUsers 432, 433
API.User.User.GetUsersForTask 433
API.User.User.GetUsersNotInGroup 433
API.User.User.IsAdmin 434
API.User.User.IsARoleMember 435, 436
API.User.User.IsARoleMemberForContent 436
API.User.User.IsARoleMemberForContent_FolderUserAdmin 437
API.User.User.IsARoleMemberForFolder 437
API.User.User.IsARoleMemberForFolder_FolderUserAdmin 438
API.User.User.IsARoleMemberForFolder_MetadataAdmin 438
API.User.User.LogInUser 438
API.User.User.ReadCustomProperties 439
API.User.User.ResetMembershipUserPassword 439
API.User.User.SetAuthenticationCookie 439
API.User.User.UnsubscribeUser 440
API.User.User.UpdateCustomPropertiesItemOrder 440
API.User.User.UpdateUser 441
API.User.User.UpDateUserGroup 441, 442
API.User.User.UpdateUserPreferences 442
API.User.User.UserObject 442, 443
API.Utilities 524
API.Utilities.LoadPermissions 525
Asset class 338, 554
- about Asset class 338, 554
- Asset methods 339, 556
- GetAssetMgtConfigInfo 340, 556
- GetAssetMgtInfo 340, 556
- GetAssetSupertypes 340, 557
- GetCreateDragDropSnippet 341, 557
- GetCreateEktronExplorerSnippet 341, 557
- GetCreateSnippet 341, 558
- GetDragDropWOSnippet 341, 558
- GetEditSnippet 342, 558

GetMyAssetMgtID 342, 559
GetViewUrl 342
MakeCopyDMSAsset 343, 559
SetAssetMgtConfigInfo 343, 560
AssetControl class 15
about AssetControl class 15
AssetControl methods 16
AssetControl properties 16
DynamicParameter 17
Fill 16
UploadType 17

B

Blog class 17, 344, 560
about Blog class 17, 344, 560
AddBlog 346, 562
AddCommentForPost 346, 563
AddPost 346, 563
Blog methods 19, 345, 562
Blog properties 20
CacheInterval 20
DeleteBlog 347, 564
DeletePost 347, 564
Fill 19
GetBlankBlogPostData 347, 564
GetBlog 348, 349, 565, 566
GetBlog method 348, 565
GetBlogData 350, 351, 567, 568, 569
GetBlogData method 349, 566
GetBlogDataForPost 352, 570
GetBlogPostData 352, 570
GetBlogRoll 353, 571
GetBlogRoll method 352, 570
GetPost 354, 572
PostParameter 20
RecentPosts 20
ShowHeader 21
ToString 19
Wrap 19
BlogArchive class 21
about BlogArchive class 21
BlogArchive methods 22
BlogArchive properties 23
CacheInterval 23
Fill 22
Text 23
Wrap 23
WrapTag 24
BlogCalendar class 24
about BlogCalendar class 24
BlogCalendar methods 25
BlogCalendar properties 26
CacheInterval 26
Fill 25
Text 26
Wrap 25
WrapTag 26
BlogCategories class 27
about BlogCategories class 27
BlogCategories methods 28
BlogCategories properties 29
CacheInterval 29
Fill 28
Text 29
Wrap 28
WrapTag 29
BlogEntries class 30
about BlogEntries class 30
BlogEntries methods 31
BlogEntries properties 33
BlogPostParameter 33
CacheInterval 34
DisplayXslt 34
Fill 31
GetRssFeed 32
GetView 32
GetViewNames 32
ShowHeader 34

ToString 32
Wrap 33
WrapTag 34
XmlDoc 35
BlogPost class 35
about BlogPost class 35
BlogPost methods 36
Wrap 36
BlogRecentPosts class 36
about BlogRecentPosts class 36
BlogRecentPosts methods 38
BlogRecentPosts properties 38
CacheInterval 39
Fill 38
NumberofPosts 39
Text 39
Wrap 38
WrapTag 39
BlogRoll class 40
about BlogRoll class 40
BlogRoll methods 41
BlogRoll properties 41
CacheInterval 42
Fill 41
Text 42
Wrap 41
WrapTag 42
BlogRSS class 42
about BlogRSS class 42
BlogRSS methods 44
BlogRSS properties 44
CacheInterval 44
Fill 44
Text 44
WrapTag 45
BreadCrumb class 45
about BreadCrumb class 45
BreadCrumb methods 46
BreadCrumb properties 48
DynamicParameter 48
Fill 47
GenerateTrail_HTML 47
MetadataName 49
ResetTrail 47, 48
ResetTrail method 47
BreadCrumbItem class 49
about BreadCrumbItem class 49
BreadCrumbItem 51
BreadCrumbItem constructor 51
BreadCrumbItem methods 51
BreadCrumbItem properties 52
CaseSensitiveUrlComparison 53
DisplayTitle 53
Equals 52
GetHashCode 52
IconAlt 53
IconPath 53
RawUrl 53
Target 54
ToString 52
BusinessRule class 444
about BusinessRule class 444
BusinessRule methods 444
GetBusinessRules 445
GetBusinessRules method 445
BusinessRules class 54, 572
about BusinessRules class 54, 572
BusinessRules methods 55, 573
BusinessRules properties 56
Fill 55
GetBusinessRules 573, 574
GetBusinessRules method 573
Parameters 56
Result 56
Text 56
Wrap 55

C

Calendar class 57, 336, 575
- about Calendar class 57, 336, 575
- Calendar methods 58
- Calendar properties 59
- DefaultCalendarID 60
- DisplayType 60
- DynamicParameter 60
- EndDate 61
- Fill 59
- HighlightEventID 61
- HighlightEventTypeID 61
- ShowXML 62
- StartDate 62
- Text 62
- Wrap 59

CalendarEvent class 337, 575
- about CalendarEvent class 337, 575

CalendarEventType class 337, 575
- about CalendarEventType class 337, 575

Classes 692

Collection class 63
- about Collection class 63
- CacheInterval 68
- Collection methods 65
- Collection properties 68
- ContentParameter 69
- CopyTo 66
- Count 69
- Current 69
- DefaultCollectionID 70
- DynamicParameter 70
- EkItems 70
- EnablePaging 71
- Fill 66
- GetEnumerator 66
- GetHtml 71
- GetRssFeed 66
- GetView 66
- GetViewNames 67
- IncludeIcons 71
- IsSynchronized 72
- LinkTarget 72
- MaxResults 72
- MoveNext 67
- Random 72
- Reset 67
- SyncRoot 73
- this 67
- ToString 67
- Wrap 68
- XmlDoc 73

CommentData 692

CommentData class 692
- about CommentData class 692

Content class 354, 576
- about Content class 354, 576
- AddContent 360, 361, 579, 580
- AddContent method 360, 579
- AddTaxonomyItem 362
- AddTaxonomyItem method 362
- AddTaxonomySyncFolder 363
- AddXmlConfiguration 363
- CheckOutContent 363, 581
- Content methods 358, 578
- CopyContentToFolder 364, 581
- CreateTaxonomy 364
- DeleteContentItem 364, 582
- DeleteTaxonomy 365
- DeleteXmlConfiguration 365
- GetAddViewLanguage 365, 582
- GetAllComments 365, 583
- GetAllTemplates 366, 583
- GetAllUnassignedItemApprovals 366, 583
- GetAllXmlConfigurations 367
- GetChildContent 367, 584
- GetContent 368, 584

GetContent method 367
GetContentByHistory 369, 584
GetContentDisplayLink 369
GetContentForEditing 370, 585
GetContentReport 370
GetContentState 370, 585
GetContentStatus 371, 585
GetDomain 371, 586
GetExpireContent 371
GetTaxonomyIdByPath 372
LoadTaxonomy 372
MoveContentToFolder 372, 586
PublishContent 373, 586
ReadAllAssignedCategory 373
ReadCategoryByTaxonomy 374
ReadTaxonomy 374
RelicateXmlConfiguration 374
RemoveTaxonomyFolder 375
RemoveTaxonomyItem 375
RenameContent 375, 587
ReOrderTaxonomyItems 375
RestoreHistoryContent 376
SaveContent 376, 587
SubmitForDelete 377, 588
UndoCheckout 377, 588
UpdateContentMetaData 377, 588
UpdateTaxonomy 378
UpdatexmlCollectionPackage 378
UpdateXmlConfiguration 378
ContentBlock class 73
about ContentBlock class 73
AllowMembershipEditing 78
CacheInterval 78
ContentBlock methods 75
ContentBlock properties 77
CopyTo 76
Count 78
Current 79
DisplayXslt 79
DynamicParameter 79
EkItem 80
Fill 76
GetEnumerator 76
IsSynchronized 80
MoveNext 76
OverrideXslt 80
Reset 76
StartingFolderID 80
SyncRoot 81
this 77
ToString 77
Wrap 77
ContentInfo class 293
about ContentInfo class 293
AssetID 295
Comments 295
ContentInfo properties 294
ErrorMessage 295
FileAndPath 295
FileExtension 295
FileName 296
FolderID 296
Html 296
ID 296
Language 296
MimeType 296
Status 297
Title 297
Type 297
UserID 297
ContentList class 81
about ContentList class 81
CacheInterval 86
ContentIds 87
ContentList methods 83
ContentList properties 86
CopyTo 84
Count 87

Current 87
DefaultContentID 87
Direction 88
DynamicParameter 88
EkItems 88
EnablePaging 89
Fill 84
GetEnumerator 84
GetHtml 89
GetRssFeed 84
IncludeIcons 89
IsSynchronized 90
LinkTarget 90
MaxResults 90
MetaTag 90
MoveNext 85
OrderKey 91
Random 91
Reset 85
SyncRoot 91
this 85
ToString 85
Wrap 86
XmlDoc 91
ContentRating class 92, 379, 589
about ContentRating class 92, 379, 589
AddContentRating 590
AlreadyRatedMessage 95
BadLabel 96
CharsRemainingLabel 96
CompleteGraphDisplay 96
ContentRating methods 94, 380, 590
ContentRating properties 95
CookiesRequiredMessage 97
DefaultContentID 97
DynamicParameter 97
FeedbackHeader 98
Fill 94
GetContentRating 380
GetContentRatingAddRating 381
GetContentRatingResults 381, 590
GetContentRatingStatistics 382
GetContentRatingStatistics method 381
GoodLabel 98
InitialGraphDisplay 98
Padding 99
PurgeContentRatings 383, 591
RatingLevelLabel 99
ThankYouMessage 99
ToString 94
TotalRatingsLabel 100
UserCommentsHeader 100
ValidationMessage 100
Wrap 94
CustomFields class 446, 591
about CustomFields class 446, 591
AddNewEditableCustomFieldAssignments 450, 594
BoolToYesNo 450, 595
CustomFields methods 449, 593
GetEditableCustomFieldAssignments 450, 595
GetEnhancedMetadataArea 451, 596
GetEnhancedMetadataScript 451, 596
GetFieldsByFolder 451, 597
GetSearchProperties 452, 453, 597
GetSearchProperties method 452
Populate_AssetRequestObjectFromForm 453
ProcessCustomFields 454, 597
SearchAllAssets 454, 598
WriteFilteredMetadataForEdit 454, 598
WriteFilteredMetadataForView 455, 599
WriteLibrarySearchExtended 455, 599
WriteMetadataDefaultForEdit 455, 599
WriteMetadataTypeForView 456, 600
WritePublicSearch 456, 600
WritePublicSearchEx 457

D

DateTimeSelector class 458

about DateTimeSelector class 458
ClientDateSeparator 462
ClientLongDatePattern 462
ClientLongTimePattern 463
ClientShortDatePattern 463
ClientShortTimePattern 463
ClientTimeSeparator 463
DateTimeSelector methods 460
DateTimeSelector properties 461
DisplayCultureDate 460
DisplayCultureDateTime 460
DisplayCultureTime 461
ExtendedMeta 464
FormElement 464
FormName 464
RequestInfo 464
RoundMinutes 461
ServerDateSeparator 465
ServerShortDatePattern 465
ServerShortTimePattern 465
ServerTimeSeparator 465
SpanAttributes 465
SpanId 466
TargetDate 466
ValidationType 466
DesignTimeDiagnostic class 101
about DesignTimeDiagnostic class 101
Authenticated 103
DesignTimeDiagnostic methods 102
DesignTimeDiagnostic properties 102
Fill 102
Hide 103
Language 103
SuppressWrapperTags 103
WrapTag 104
DhtmlMenu class 104
about DhtmlMenu class 104
AccessKey 108
Align 108
BackColor 108
BorderColor 108
BorderStyle 109
BorderWidth 109
CssClass 109
DhtmlMenu 106
DhtmlMenu methods 106
DhtmlMenu properties 107
DisplayXslt 109
Enabled 109
EnableIE6FormsFix 110
Fill 106
Height 110
HorizontalDisplay 110
MenuHoverOverColor 110
MenuStyle 111
Stylesheet 111
SubMenuBackColor 111
SubMenuHoverOverColor 112
SubMenuItemHeight 112
SubMenuItemWidth 112
SubMenuTopOffset 113
TabIndex 113
ToolTip 113
Width 113
Wrap 107

E

EkEditorPart class 7
about EkEditorPart class 7
Ektron 5
Ektron namespace 5
Ektron.Cms 5
Ektron.Cms namespace 5
Ektron.Cms.Controls 5
Ektron.Cms.Controls namespace 5
Classes 8
Ektron.Cms.Controls.ActiveTopics 9
Ektron.Cms.Controls.ActiveTopics.BoardID 12

Ektron.Cms.Controls.ActiveTopics.CacheInterval 12
Ektron.Cms.Controls.ActiveTopics.Fill 11
Ektron.Cms.Controls.ActiveTopics.ToString 11
Ektron.Cms.Controls.ActiveTopics.Wrap 11
Ektron.Cms.Controls.AnalyticsTracker 12
Ektron.Cms.Controls.AnalyticsTracker.DefaultContentID 14
Ektron.Cms.Controls.AnalyticsTracker.DynamicParameter 14
Ektron.Cms.Controls.AnalyticsTracker.EnableAnalytics 15
Ektron.Cms.Controls.AnalyticsTracker.Wrap 14
Ektron.Cms.Controls.AssetControl 15
Ektron.Cms.Controls.AssetControl.DynamicParameter 17
Ektron.Cms.Controls.AssetControl.Fill 16
Ektron.Cms.Controls.AssetControl.UploadType 17
Ektron.Cms.Controls.Blog 17
Ektron.Cms.Controls.Blog.CacheInterval 20
Ektron.Cms.Controls.Blog.Fill 19
Ektron.Cms.Controls.Blog.PostParameter 20
Ektron.Cms.Controls.Blog.RecentPosts 20
Ektron.Cms.Controls.Blog.ShowHeader 21
Ektron.Cms.Controls.Blog.ToString 19
Ektron.Cms.Controls.Blog.Wrap 19
Ektron.Cms.Controls.BlogArchive 21
Ektron.Cms.Controls.BlogArchive.CacheInterval 23
Ektron.Cms.Controls.BlogArchive.Fill 22
Ektron.Cms.Controls.BlogArchive.Text 23
Ektron.Cms.Controls.BlogArchive.Wrap 23
Ektron.Cms.Controls.BlogArchive.WrapTag 24
Ektron.Cms.Controls.BlogCalendar 24
Ektron.Cms.Controls.BlogCalendar.CacheInterval 26
Ektron.Cms.Controls.BlogCalendar.Fill 25
Ektron.Cms.Controls.BlogCalendar.Text 26
Ektron.Cms.Controls.BlogCalendar.Wrap 25
Ektron.Cms.Controls.BlogCalendar.WrapTag 26
Ektron.Cms.Controls.BlogCategories 27
Ektron.Cms.Controls.BlogCategories.CacheInterval 29
Ektron.Cms.Controls.BlogCategories.Fill 28
Ektron.Cms.Controls.BlogCategories.Text 29
Ektron.Cms.Controls.BlogCategories.Wrap 28
Ektron.Cms.Controls.BlogCategories.WrapTag 29
Ektron.Cms.Controls.BlogEntries 30
Ektron.Cms.Controls.BlogEntries.BlogPostParameter 33
Ektron.Cms.Controls.BlogEntries.CacheInterval 34
Ektron.Cms.Controls.BlogEntries.DisplayXslt 34
Ektron.Cms.Controls.BlogEntries.Fill 31
Ektron.Cms.Controls.BlogEntries.GetRssFeed 32
Ektron.Cms.Controls.BlogEntries.GetView 32
Ektron.Cms.Controls.BlogEntries.GetViewNames 32
Ektron.Cms.Controls.BlogEntries.ShowHeader 34
Ektron.Cms.Controls.BlogEntries.ToString 32
Ektron.Cms.Controls.BlogEntries.Wrap 33
Ektron.Cms.Controls.BlogEntries.WrapTag 34
Ektron.Cms.Controls.BlogEntries.XmlDoc 35
Ektron.Cms.Controls.BlogPost 35
Ektron.Cms.Controls.BlogPost.Wrap 36
Ektron.Cms.Controls.BlogRecentPosts 36
Ektron.Cms.Controls.BlogRecentPosts.CacheInterval 39
Ektron.Cms.Controls.BlogRecentPosts.Fill 38
Ektron.Cms.Controls.BlogRecentPosts.NumberofPosts 39
Ektron.Cms.Controls.BlogRecentPosts.Text 39
Ektron.Cms.Controls.BlogRecentPosts.Wrap 38
Ektron.Cms.Controls.BlogRecentPosts.WrapTag 39
Ektron.Cms.Controls.BlogRoll 40
Ektron.Cms.Controls.BlogRoll.CacheInterval 42
Ektron.Cms.Controls.BlogRoll.Fill 41
Ektron.Cms.Controls.BlogRoll.Text 42
Ektron.Cms.Controls.BlogRoll.Wrap 41
Ektron.Cms.Controls.BlogRoll.WrapTag 42
Ektron.Cms.Controls.BlogRSS 42
Ektron.Cms.Controls.BlogRSS.CacheInterval 44
Ektron.Cms.Controls.BlogRSS.Fill 44
Ektron.Cms.Controls.BlogRSS.Text 44
Ektron.Cms.Controls.BlogRSS.WrapTag 45
Ektron.Cms.Controls.BreadCrumb 45
Ektron.Cms.Controls.BreadCrumb.DynamicParameter 48
Ektron.Cms.Controls.BreadCrumb.Fill 47
Ektron.Cms.Controls.BreadCrumb.GenerateTrail_HTML 47
Ektron.Cms.Controls.BreadCrumb.MetadataName 49
Ektron.Cms.Controls.BreadCrumb.ResetTrail 47, 48

Ektron.Cms.Controls.BreadCrumbItem 49
Ektron.Cms.Controls.BreadCrumbItem.BreadCrumbItem 51
Ektron.Cms.Controls.BreadCrumbItem.CaseSensitiveUrlComparison 53
Ektron.Cms.Controls.BreadCrumbItem.DisplayTitle 53
Ektron.Cms.Controls.BreadCrumbItem.Equals 52
Ektron.Cms.Controls.BreadCrumbItem.GetHashCode 52
Ektron.Cms.Controls.BreadCrumbItem.IconAlt 53
Ektron.Cms.Controls.BreadCrumbItem.IconPath 53
Ektron.Cms.Controls.BreadCrumbItem.RawUrl 53
Ektron.Cms.Controls.BreadCrumbItem.Target 54
Ektron.Cms.Controls.BreadCrumbItem.ToString 52
Ektron.Cms.Controls.BusinessRules 54
Ektron.Cms.Controls.BusinessRules.Fill 55
Ektron.Cms.Controls.BusinessRules.Parameters 56
Ektron.Cms.Controls.BusinessRules.Result 56
Ektron.Cms.Controls.BusinessRules.Text 56
Ektron.Cms.Controls.BusinessRules.Wrap 55
Ektron.Cms.Controls.Calendar 57
Ektron.Cms.Controls.Calendar.DefaultCalendarID 60
Ektron.Cms.Controls.Calendar.DisplayType 60
Ektron.Cms.Controls.Calendar.DynamicParameter 60
Ektron.Cms.Controls.Calendar.EndDate 61
Ektron.Cms.Controls.Calendar.Fill 59
Ektron.Cms.Controls.Calendar.HighlightEventID 61
Ektron.Cms.Controls.Calendar.HighlightEventTypeID 61
Ektron.Cms.Controls.Calendar.ShowXML 62
Ektron.Cms.Controls.Calendar.StartDate 62
Ektron.Cms.Controls.Calendar.Text 62
Ektron.Cms.Controls.Calendar.Wrap 59
Ektron.Cms.Controls.Collection 63
Ektron.Cms.Controls.Collection.CacheInterval 68
Ektron.Cms.Controls.Collection.ContentParameter 69
Ektron.Cms.Controls.Collection.CopyTo 66
Ektron.Cms.Controls.Collection.Count 69
Ektron.Cms.Controls.Collection.Current 69
Ektron.Cms.Controls.Collection.DefaultCollectionID 70
Ektron.Cms.Controls.Collection.DynamicParameter 70
Ektron.Cms.Controls.Collection.EkItems 70
Ektron.Cms.Controls.Collection.EnablePaging 71
Ektron.Cms.Controls.Collection.Fill 66
Ektron.Cms.Controls.Collection.GetEnumerator 66
Ektron.Cms.Controls.Collection.GetHtml 71
Ektron.Cms.Controls.Collection.GetRssFeed 66
Ektron.Cms.Controls.Collection.GetView 66
Ektron.Cms.Controls.Collection.GetViewNames 67
Ektron.Cms.Controls.Collection.IncludeIcons 71
Ektron.Cms.Controls.Collection.IsSynchronized 72
Ektron.Cms.Controls.Collection.LinkTarget 72
Ektron.Cms.Controls.Collection.MaxResults 72
Ektron.Cms.Controls.Collection.MoveNext 67
Ektron.Cms.Controls.Collection.Random 72
Ektron.Cms.Controls.Collection.Reset 67
Ektron.Cms.Controls.Collection.SyncRoot 73
Ektron.Cms.Controls.Collection.this 67
Ektron.Cms.Controls.Collection.ToString 67
Ektron.Cms.Controls.Collection.Wrap 68
Ektron.Cms.Controls.Collection.XmlDoc 73
Ektron.Cms.Controls.ContentBlock 73
Ektron.Cms.Controls.ContentBlock.AllowMembershipEditing 78
Ektron.Cms.Controls.ContentBlock.CacheInterval 78
Ektron.Cms.Controls.ContentBlock.CopyTo 76
Ektron.Cms.Controls.ContentBlock.Count 78
Ektron.Cms.Controls.ContentBlock.Current 79
Ektron.Cms.Controls.ContentBlock.DisplayXslt 79
Ektron.Cms.Controls.ContentBlock.DynamicParameter 79
Ektron.Cms.Controls.ContentBlock.EkItem 80
Ektron.Cms.Controls.ContentBlock.Fill 76
Ektron.Cms.Controls.ContentBlock.GetEnumerator 76
Ektron.Cms.Controls.ContentBlock.IsSynchronized 80
Ektron.Cms.Controls.ContentBlock.MoveNext 76
Ektron.Cms.Controls.ContentBlock.OverrideXslt 80
Ektron.Cms.Controls.ContentBlock.Reset 76
Ektron.Cms.Controls.ContentBlock.StartingFolderID 80
Ektron.Cms.Controls.ContentBlock.SyncRoot 81
Ektron.Cms.Controls.ContentBlock.this 77
Ektron.Cms.Controls.ContentBlock.ToString 77
Ektron.Cms.Controls.ContentBlock.Wrap 77

Ektron.Cms.Controls.ContentList 81
Ektron.Cms.Controls.ContentList.CacheInterval 86
Ektron.Cms.Controls.ContentList.ContentIds 87
Ektron.Cms.Controls.ContentList.CopyTo 84
Ektron.Cms.Controls.ContentList.Count 87
Ektron.Cms.Controls.ContentList.Current 87
Ektron.Cms.Controls.ContentList.DefaultContentID 87
Ektron.Cms.Controls.ContentList.Direction 88
Ektron.Cms.Controls.ContentList.DynamicParameter 88
Ektron.Cms.Controls.ContentList.EkItems 88
Ektron.Cms.Controls.ContentList.EnablePaging 89
Ektron.Cms.Controls.ContentList.Fill 84
Ektron.Cms.Controls.ContentList.GetEnumerator 84
Ektron.Cms.Controls.ContentList.GetHtml 89
Ektron.Cms.Controls.ContentList.GetRssFeed 84
Ektron.Cms.Controls.ContentList.IncludeIcons 89
Ektron.Cms.Controls.ContentList.IsSynchronized 90
Ektron.Cms.Controls.ContentList.LinkTarget 90
Ektron.Cms.Controls.ContentList.MaxResults 90
Ektron.Cms.Controls.ContentList.MetaTag 90
Ektron.Cms.Controls.ContentList.MoveNext 85
Ektron.Cms.Controls.ContentList.OrderKey 91
Ektron.Cms.Controls.ContentList.Random 91
Ektron.Cms.Controls.ContentList.Reset 85
Ektron.Cms.Controls.ContentList.SyncRoot 91
Ektron.Cms.Controls.ContentList.this 85
Ektron.Cms.Controls.ContentList.ToString 85
Ektron.Cms.Controls.ContentList.Wrap 86
Ektron.Cms.Controls.ContentList.XmlDoc 91
Ektron.Cms.Controls.ContentRating 92
Ektron.Cms.Controls.ContentRating.AlreadyRatedMessage 95
Ektron.Cms.Controls.ContentRating.BadLabel 96
Ektron.Cms.Controls.ContentRating.CharsRemainingLabel 96
Ektron.Cms.Controls.ContentRating.CompleteGraphDisplay 96
Ektron.Cms.Controls.ContentRating.CookiesRequiredMessage 97
Ektron.Cms.Controls.ContentRating.DefaultContentID 97
Ektron.Cms.Controls.ContentRating.DynamicParameter 97
Ektron.Cms.Controls.ContentRating.FeedbackHeader 98
Ektron.Cms.Controls.ContentRating.Fill 94
Ektron.Cms.Controls.ContentRating.GoodLabel 98
Ektron.Cms.Controls.ContentRating.InitialGraphDisplay 98
Ektron.Cms.Controls.ContentRating.Padding 99
Ektron.Cms.Controls.ContentRating.RatingLevelLabel 99
Ektron.Cms.Controls.ContentRating.ThankYouMessage 99
Ektron.Cms.Controls.ContentRating.ToString 94
Ektron.Cms.Controls.ContentRating.TotalRatingsLabel 100
Ektron.Cms.Controls.ContentRating.UserCommentsHeader 100
Ektron.Cms.Controls.ContentRating.ValidationMessage 100
Ektron.Cms.Controls.ContentRating.Wrap 94
Ektron.Cms.Controls.DesignTimeDiagnostic 101
Ektron.Cms.Controls.DesignTimeDiagnostic.Authenticated 103
Ektron.Cms.Controls.DesignTimeDiagnostic.Fill 102
Ektron.Cms.Controls.DesignTimeDiagnostic.Hide 103
Ektron.Cms.Controls.DesignTimeDiagnostic.Language 103
Ektron.Cms.Controls.DesignTimeDiagnostic.SuppressWrapperTags 103
Ektron.Cms.Controls.DesignTimeDiagnostic.WrapTag 104
Ektron.Cms.Controls.DhtmlMenu 104
Ektron.Cms.Controls.DhtmlMenu.AccessKey 108
Ektron.Cms.Controls.DhtmlMenu.Align 108
Ektron.Cms.Controls.DhtmlMenu.BackColor 108
Ektron.Cms.Controls.DhtmlMenu.BorderColor 108
Ektron.Cms.Controls.DhtmlMenu.BorderStyle 109
Ektron.Cms.Controls.DhtmlMenu.BorderWidth 109
Ektron.Cms.Controls.DhtmlMenu.CssClass 109
Ektron.Cms.Controls.DhtmlMenu.DhtmlMenu 106
Ektron.Cms.Controls.DhtmlMenu.DisplayXslt 109
Ektron.Cms.Controls.DhtmlMenu.Enabled 109
Ektron.Cms.Controls.DhtmlMenu.EnableIE6FormsFix 110
Ektron.Cms.Controls.DhtmlMenu.Fill 106
Ektron.Cms.Controls.DhtmlMenu.Height 110
Ektron.Cms.Controls.DhtmlMenu.HorizontalDisplay 110
Ektron.Cms.Controls.DhtmlMenu.MenuHoverOverColor 110
Ektron.Cms.Controls.DhtmlMenu.MenuStyle 111
Ektron.Cms.Controls.DhtmlMenu.Stylesheet 111
Ektron.Cms.Controls.DhtmlMenu.SubMenuBackColor 111
Ektron.Cms.Controls.DhtmlMenu.SubMenuHoverOverColor 112
Ektron.Cms.Controls.DhtmlMenu.SubMenuItemHeight 112

Ektron.Cms.Controls.DhtmlMenu.SubMenuItemWidth 112
Ektron.Cms.Controls.DhtmlMenu.SubMenuTopOffset 113
Ektron.Cms.Controls.DhtmlMenu.TabIndex 113
Ektron.Cms.Controls.DhtmlMenu.ToolTip 113
Ektron.Cms.Controls.DhtmlMenu.Width 113
Ektron.Cms.Controls.DhtmlMenu.Wrap 107
Ektron.Cms.Controls.EktronCatalogPart 114
Ektron.Cms.Controls.EktronCatalogPart.DefaultContentID 115
Ektron.Cms.Controls.EktronCommunityCatalogPart 115
Ektron.Cms.Controls.EktronWebPartZone 115
Ektron.Cms.Controls.EktronWebPartZone.CmsSitePath 116
Ektron.Cms.Controls.EkWebControl 117
Ektron.Cms.Controls.EkWebControl.Authenticated 122
Ektron.Cms.Controls.EkWebControl.EkWebControl 119
Ektron.Cms.Controls.EkWebControl.Error 122
Ektron.Cms.Controls.EkWebControl.Fill 119
Ektron.Cms.Controls.EkWebControl.Hide 122
Ektron.Cms.Controls.EkWebControl.IsLoggedIn 123
Ektron.Cms.Controls.EkWebControl.Language 123
Ektron.Cms.Controls.EkWebControl.LanguageID 123
Ektron.Cms.Controls.EkWebControl.LocalizeString 120
Ektron.Cms.Controls.EkWebControl.loggedInUserID 123
Ektron.Cms.Controls.EkWebControl.loggedInUserName 124
Ektron.Cms.Controls.EkWebControl.RegisterClientScriptInclude 120
Ektron.Cms.Controls.EkWebControl.RegisterStylesheet 120
Ektron.Cms.Controls.EkWebControl.SeachHeader 120
Ektron.Cms.Controls.EkWebControl.SuppressWrapperTags 124
Ektron.Cms.Controls.EkWebControl.Text 124
Ektron.Cms.Controls.EkWebControl.ToString 121
Ektron.Cms.Controls.EkWebControl.Wrap 121
Ektron.Cms.Controls.EkWebControl.WrapTag 124
Ektron.Cms.Controls.EkWebPart 125
Ektron.Cms.Controls.EkXsltBase 125
Ektron.Cms.Controls.EkXsltBase.DisplayXslt 127
Ektron.Cms.Controls.EkXsltBase.EkXsltBase 126
Ektron.Cms.Controls.EkXsltBase.SetDirty 127
Ektron.Cms.Controls.EkXsltBase.XmlDoc 128
Ektron.Cms.Controls.EkXsltBase.XsltDoc 128
Ektron.Cms.Controls.EkXsltWebPart 128
Ektron.Cms.Controls.FolderBreadcrumb 128
Ektron.Cms.Controls.FolderBreadcrumb.DynamicParameter 130
Ektron.Cms.Controls.FolderBreadcrumb.Fill 130
Ektron.Cms.Controls.FormBlock 130
Ektron.Cms.Controls.FormBlock.AddValidation 136
Ektron.Cms.Controls.FormBlock.CacheInterval 136
Ektron.Cms.Controls.FormBlock.CopyTo 133
Ektron.Cms.Controls.FormBlock.Count 136
Ektron.Cms.Controls.FormBlock.Current 136
Ektron.Cms.Controls.FormBlock.DynamicParameter 137
Ektron.Cms.Controls.FormBlock.EkItem 137
Ektron.Cms.Controls.FormBlock.Fields 137
Ektron.Cms.Controls.FormBlock.Fill 133
Ektron.Cms.Controls.FormBlock.FormName 137
Ektron.Cms.Controls.FormBlock.GetEnumerator 133
Ektron.Cms.Controls.FormBlock.GetView 133
Ektron.Cms.Controls.FormBlock.GetViewNames 134
Ektron.Cms.Controls.FormBlock.IncludeTags 138
Ektron.Cms.Controls.FormBlock.IsSynchronized 138
Ektron.Cms.Controls.FormBlock.MoveNext 134
Ektron.Cms.Controls.FormBlock.Reset 134
Ektron.Cms.Controls.FormBlock.StartingFolderID 138
Ektron.Cms.Controls.FormBlock.SyncRoot 139
Ektron.Cms.Controls.FormBlock.this 134
Ektron.Cms.Controls.FormBlock.ToString 135
Ektron.Cms.Controls.FormBlock.Wrap 135
Ektron.Cms.Controls.FormFieldItem 139
Ektron.Cms.Controls.FormFieldItem.Basetype 140
Ektron.Cms.Controls.FormFieldItem.ContentType 140
Ektron.Cms.Controls.FormFieldItem.DatalistItems 141
Ektron.Cms.Controls.FormFieldItem.Datatype 141
Ektron.Cms.Controls.FormFieldItem.DisplayName 141
Ektron.Cms.Controls.FormFieldItem.FieldName 142
Ektron.Cms.Controls.FormFieldItem.Language 142
Ektron.Cms.Controls.FormFieldItem.Value 142
Ektron.Cms.Controls.FormFieldItemDatalistItem 142
Ektron.Cms.Controls.FormFieldItemDatalistItem.DataValue 143
Ektron.Cms.Controls.FormFieldItemDatalistItem.DisplayValue 144
Ektron.Cms.Controls.Forum 144

Ektron.Cms.Controls.Forum.CacheInterval 146
Ektron.Cms.Controls.Forum.Fill 145
Ektron.Cms.Controls.Forum.ToString 145
Ektron.Cms.Controls.Forum.Wrap 146
Ektron.Cms.Controls.HtmlEditDisplay 147
Ektron.Cms.Controls.HtmlEditDisplay.BaseJavaScript 148
Ektron.Cms.Controls.HtmlEditDisplay.BeginJavaScript 149
Ektron.Cms.Controls.HtmlEditDisplay.EndJavaScript 149
Ektron.Cms.Controls.HtmlEditDisplay.MakeFieldNameFromId 149
Ektron.Cms.Controls.HtmlEditDisplay.RenderCMSFields 150
Ektron.Cms.Controls.HtmlEditDisplay.SelectedJavaScript 150
Ektron.Cms.Controls.HtmlEditor 150
Ektron.Cms.Controls.HtmlEditor.Archived 157
Ektron.Cms.Controls.HtmlEditor.Authenticated 157
Ektron.Cms.Controls.HtmlEditor.CmsSitePath 158
Ektron.Cms.Controls.HtmlEditor.ContentAction 158
Ektron.Cms.Controls.HtmlEditor.ContentTitle 159
Ektron.Cms.Controls.HtmlEditor.ContentType 159
Ektron.Cms.Controls.HtmlEditor.EditorReadyCall 160
Ektron.Cms.Controls.HtmlEditor.EnableUpload 160
Ektron.Cms.Controls.HtmlEditor.ExecuteContentAction 155
Ektron.Cms.Controls.HtmlEditor.HtmlEditor 154
Ektron.Cms.Controls.HtmlEditor.IncludeContentInfo 161
Ektron.Cms.Controls.HtmlEditor.IsLoggedIn 161
Ektron.Cms.Controls.HtmlEditor.loggedInUserID 162
Ektron.Cms.Controls.HtmlEditor.loggedInUserName 162
Ektron.Cms.Controls.HtmlEditor.Path 162
Ektron.Cms.Controls.HtmlEditor.PreferredType 163
Ektron.Cms.Controls.HtmlEditor.SafeEncode 163
Ektron.Cms.Controls.HtmlEditor.ShowCMSToolbar 164
Ektron.Cms.Controls.HtmlEditor.ShowTitle 164
Ektron.Cms.Controls.HtmlEditor.StyleSheet 165
Ektron.Cms.Controls.HtmlEditor.Text 165
Ektron.Cms.Controls.HtmlEditor.ToolbarLevel 166
Ektron.Cms.Controls.HtmlEditor.ToolbarResetCall 166
Ektron.Cms.Controls.HtmlEditor.ToString 155
Ektron.Cms.Controls.HtmlEditor.TranslationFile 167
Ektron.Cms.Controls.HtmlEditor.WorkareaMode 155
Ektron.Cms.Controls.IndexSearch 167
Ektron.Cms.Controls.IndexSearch.AddParm 173
Ektron.Cms.Controls.IndexSearch.ButtonImgSrc 177
Ektron.Cms.Controls.IndexSearch.ButtonText 178
Ektron.Cms.Controls.IndexSearch.ContentParameter 178
Ektron.Cms.Controls.IndexSearch.CopyTo 173
Ektron.Cms.Controls.IndexSearch.Count 178
Ektron.Cms.Controls.IndexSearch.Current 178
Ektron.Cms.Controls.IndexSearch.EkItems 179
Ektron.Cms.Controls.IndexSearch.EmptyResultMsg 179
Ektron.Cms.Controls.IndexSearch.EncodeContentHtml 179
Ektron.Cms.Controls.IndexSearch.Fill 173
Ektron.Cms.Controls.IndexSearch.FolderId 179
Ektron.Cms.Controls.IndexSearch.FormName 180
Ektron.Cms.Controls.IndexSearch.GetEnumerator 174
Ektron.Cms.Controls.IndexSearch.GetSearchParmDataType 174
Ektron.Cms.Controls.IndexSearch.GetSearchParmSearchType 174
Ektron.Cms.Controls.IndexSearch.GetXPathValues 174
Ektron.Cms.Controls.IndexSearch.HitsMarker 180
Ektron.Cms.Controls.IndexSearch.IndexSearch 171
Ektron.Cms.Controls.IndexSearch.IsSynchronized 180
Ektron.Cms.Controls.IndexSearch.Labelafter 180
Ektron.Cms.Controls.IndexSearch.Labelbefore 181
Ektron.Cms.Controls.IndexSearch.LabelBetween 181
Ektron.Cms.Controls.IndexSearch.LabelContains 181
Ektron.Cms.Controls.IndexSearch.LabeldateBetween 182
Ektron.Cms.Controls.IndexSearch.LabelEqual 182
Ektron.Cms.Controls.IndexSearch.LabelExactPhrase 182
Ektron.Cms.Controls.IndexSearch.LabelGreaterThan 183
Ektron.Cms.Controls.IndexSearch.Labelinputdate 183
Ektron.Cms.Controls.IndexSearch.Labelinputnumber 183
Ektron.Cms.Controls.IndexSearch.LabelLessThan 184
Ektron.Cms.Controls.IndexSearch.Labelon 184
Ektron.Cms.Controls.IndexSearch.LabelSelect 184
Ektron.Cms.Controls.IndexSearch.LinkTarget 185
Ektron.Cms.Controls.IndexSearch.MaxResults 185
Ektron.Cms.Controls.IndexSearch.MoveNext 175
Ektron.Cms.Controls.IndexSearch.OrderBy 185
Ektron.Cms.Controls.IndexSearch.OrderByDirection 185

Ektron.Cms.Controls.IndexSearch.Recursive 186
Ektron.Cms.Controls.IndexSearch.Reset 175
Ektron.Cms.Controls.IndexSearch.Search 175
Ektron.Cms.Controls.IndexSearch.SearchParmXML 186
Ektron.Cms.Controls.IndexSearch.SearchPram 172
Ektron.Cms.Controls.IndexSearch.SearchPram structure 172
Ektron.Cms.Controls.IndexSearch.ShowSearchBoxAlways 186
Ektron.Cms.Controls.IndexSearch.SyncRoot 186
Ektron.Cms.Controls.IndexSearch.this 176
Ektron.Cms.Controls.IndexSearch.Weighted 187
Ektron.Cms.Controls.IndexSearch.Wrap 176
Ektron.Cms.Controls.IndexSearch.XmlConfigId 187
Ektron.Cms.Controls.IndexSearch.XmlDoc 187
Ektron.Cms.Controls.LanguageAPI 187
Ektron.Cms.Controls.LanguageAPI.CurrentLanguage 191
Ektron.Cms.Controls.LanguageAPI.CurrentLanguageID 192
Ektron.Cms.Controls.LanguageAPI.DefaultLanguage 192
Ektron.Cms.Controls.LanguageAPI.DefaultLanguageID 192
Ektron.Cms.Controls.LanguageAPI.Fill 190
Ektron.Cms.Controls.LanguageAPI.GetLanguage 190
Ektron.Cms.Controls.LanguageAPI.GetLanguageID 190
Ektron.Cms.Controls.LanguageAPI.IsValid 191
Ektron.Cms.Controls.LanguageAPI.LangaugeIdList 192
Ektron.Cms.Controls.LanguageAPI.LangaugeTitleList 192
Ektron.Cms.Controls.LanguageAPI.LanguageAPI 189
Ektron.Cms.Controls.LanguageAPI.MultiLanguageEnabled 193
Ektron.Cms.Controls.LanguageAPI.OverrideDefault 193
Ektron.Cms.Controls.LanguageAPI.SiteLanguage 193
Ektron.Cms.Controls.LanguageAPI.SiteLanguageID 193
Ektron.Cms.Controls.LanguageSelect 194
Ektron.Cms.Controls.LanguageSelect.Fill 195
Ektron.Cms.Controls.LanguageSelect.LabelName 196
Ektron.Cms.Controls.LanguageSelect.Text 196
Ektron.Cms.Controls.LanguageSelect.Wrap 195
Ektron.Cms.Controls.ListSummary 196
Ektron.Cms.Controls.ListSummary.CacheInterval 202
Ektron.Cms.Controls.ListSummary.ContentParameter 203
Ektron.Cms.Controls.ListSummary.ContentType 203
Ektron.Cms.Controls.ListSummary.CopyTo 199
Ektron.Cms.Controls.ListSummary.Count 203
Ektron.Cms.Controls.ListSummary.Current 203
Ektron.Cms.Controls.ListSummary.Direction 204
Ektron.Cms.Controls.ListSummary.EkItems 204
Ektron.Cms.Controls.ListSummary.EnablePaging 204
Ektron.Cms.Controls.ListSummary.Fill 199
Ektron.Cms.Controls.ListSummary.FolderID 205
Ektron.Cms.Controls.ListSummary.FolderPath 205
Ektron.Cms.Controls.ListSummary.GetEnumerator 199
Ektron.Cms.Controls.ListSummary.GetHtml 205
Ektron.Cms.Controls.ListSummary.GetRssFeed 200
Ektron.Cms.Controls.ListSummary.GetView 200
Ektron.Cms.Controls.ListSummary.GetViewNames 200
Ektron.Cms.Controls.ListSummary.IncludeIcons 206
Ektron.Cms.Controls.ListSummary.IsSynchronized 206
Ektron.Cms.Controls.ListSummary.LinkTarget 206
Ektron.Cms.Controls.ListSummary.MaxResults 207
Ektron.Cms.Controls.ListSummary.MoveNext 200
Ektron.Cms.Controls.ListSummary.OrderKey 207
Ektron.Cms.Controls.ListSummary.Random 207
Ektron.Cms.Controls.ListSummary.Recursive 208
Ektron.Cms.Controls.ListSummary.Reset 201
Ektron.Cms.Controls.ListSummary.SyncRoot 208
Ektron.Cms.Controls.ListSummary.this 201
Ektron.Cms.Controls.ListSummary.ToString 201
Ektron.Cms.Controls.ListSummary.Wrap 201
Ektron.Cms.Controls.ListSummary.XmlDoc 208
Ektron.Cms.Controls.Login 208
Ektron.Cms.Controls.Login.AutoAddType 210
Ektron.Cms.Controls.Login.AutoLogin 211
Ektron.Cms.Controls.Login.Fill 210
Ektron.Cms.Controls.Login.OnlyAllowMemberLogin 211
Ektron.Cms.Controls.Login.Password 211
Ektron.Cms.Controls.Login.PromptLogout 211
Ektron.Cms.Controls.Login.SuppressHelpButton 211
Ektron.Cms.Controls.Login.Text 212
Ektron.Cms.Controls.Login.UserName 212
Ektron.Cms.Controls.Membership 212
Ektron.Cms.Controls.Membership.BoardID 215

Ektron.Cms.Controls.Membership.DisplayMode 216
Ektron.Cms.Controls.Membership.Fill 215
Ektron.Cms.Controls.Membership.IsPostBack 214
Ektron.Cms.Controls.Membership.RedirectFailedURL 216
Ektron.Cms.Controls.Membership.RedirectSuccessURL 216
Ektron.Cms.Controls.Membership.RegisterButtonImg 216
Ektron.Cms.Controls.Membership.RegisterButtonText 217
Ektron.Cms.Controls.Membership.ResetButtonImg 217
Ektron.Cms.Controls.Membership.ResetButtonText 217
Ektron.Cms.Controls.Membership.ShowExtended 217
Ektron.Cms.Controls.Membership.ShowTerms 218
Ektron.Cms.Controls.Membership.Text 218
Ektron.Cms.Controls.Membership.UserExistsMessage 218
Ektron.Cms.Controls.Membership.UserID 218
Ektron.Cms.Controls.Membership.UserSuccessMessage 218
Ektron.Cms.Controls.Membership.UserUpdateSuccessMessage 219
Ektron.Cms.Controls.Menu 219
Ektron.Cms.Controls.Menu.CacheInterval 221
Ektron.Cms.Controls.Menu.DefaultMenuID 222
Ektron.Cms.Controls.Menu.DisplayXslt 222
Ektron.Cms.Controls.Menu.DynamicParameter 222
Ektron.Cms.Controls.Menu.Fill 221
Ektron.Cms.Controls.Menu.Stylesheet 223
Ektron.Cms.Controls.Menu.Wrap 221
Ektron.Cms.Controls.MetaData 223
Ektron.Cms.Controls.MetaData.DefaultContentID 224
Ektron.Cms.Controls.MetaData.Fill 224
Ektron.Cms.Controls.MetaDataList 225
Ektron.Cms.Controls.MetaDataList.CacheInterval 230
Ektron.Cms.Controls.MetaDataList.ContentType 230
Ektron.Cms.Controls.MetaDataList.CopyTo 228
Ektron.Cms.Controls.MetaDataList.Count 230
Ektron.Cms.Controls.MetaDataList.Current 230
Ektron.Cms.Controls.MetaDataList.EkItems 231
Ektron.Cms.Controls.MetaDataList.EnablePaging 231
Ektron.Cms.Controls.MetaDataList.ExactPhrase 231
Ektron.Cms.Controls.MetaDataList.Fill 228
Ektron.Cms.Controls.MetaDataList.FolderID 232
Ektron.Cms.Controls.MetaDataList.GetEnumerator 228
Ektron.Cms.Controls.MetaDataList.GetHtml 232
Ektron.Cms.Controls.MetaDataList.IncludeIcons 232
Ektron.Cms.Controls.MetaDataList.IsSynchronized 233
Ektron.Cms.Controls.MetaDataList.KeyWordName 233
Ektron.Cms.Controls.MetaDataList.KeyWordValue 233
Ektron.Cms.Controls.MetaDataList.LinkTarget 233
Ektron.Cms.Controls.MetaDataList.MaxNumber 234
Ektron.Cms.Controls.MetaDataList.MoveNext 228
Ektron.Cms.Controls.MetaDataList.OrderBy 234
Ektron.Cms.Controls.MetaDataList.Recursive 234
Ektron.Cms.Controls.MetaDataList.Reset 228
Ektron.Cms.Controls.MetaDataList.SortOrder 235
Ektron.Cms.Controls.MetaDataList.SyncRoot 235
Ektron.Cms.Controls.MetaDataList.this 229
Ektron.Cms.Controls.MetaDataList.Wrap 229
Ektron.Cms.Controls.MetaDataList.XmlDoc 235
Ektron.Cms.Controls.Personalization 7
Ektron.Cms.Controls.Personalization namespace 7
Classes 7
Ektron.Cms.Controls.Personalization.EkEditorPart 7
Ektron.Cms.Controls.PersonalizationManager 235
Ektron.Cms.Controls.PersonalizationManager.AddContentExitText 238
Ektron.Cms.Controls.PersonalizationManager.AddContentText 238
Ektron.Cms.Controls.PersonalizationManager.CurrentUserButtonText 238
Ektron.Cms.Controls.PersonalizationManager.Display 239
Ektron.Cms.Controls.PersonalizationManager.Fill 237
Ektron.Cms.Controls.PersonalizationManager.HeaderBackColor 239
Ektron.Cms.Controls.PersonalizationManager.HeaderText 239
Ektron.Cms.Controls.PersonalizationManager.PersonalizeExitText 240
Ektron.Cms.Controls.PersonalizationManager.PersonalizeScopeText 240
Ektron.Cms.Controls.PersonalizationManager.PersonalizeStartText

240

Ektron.Cms.Controls.PersonalizationManager.PublicUserButtonText
241

Ektron.Cms.Controls.PersonalizationManager.ResetLinkText
241

Ektron.Cms.Controls.PersonalizationManager.ResetLinkToolTip
242

Ektron.Cms.Controls.Poll 242

Ektron.Cms.Controls.Poll.EnableAjax 244

Ektron.Cms.Controls.Poll.Fill 243

Ektron.Cms.Controls.PostHistory 244

Ektron.Cms.Controls.PostHistory.BoardID 247

Ektron.Cms.Controls.PostHistory.CacheInterval 247

Ektron.Cms.Controls.PostHistory.Fill 246

Ektron.Cms.Controls.PostHistory.ToString 246

Ektron.Cms.Controls.PostHistory.Wrap 246

Ektron.Cms.Controls.RssAggregator 247

Ektron.Cms.Controls.RssAggregator.CopyTo 249

Ektron.Cms.Controls.RssAggregator.Count 251

Ektron.Cms.Controls.RssAggregator.Current 252

Ektron.Cms.Controls.RssAggregator.EkItems 252

Ektron.Cms.Controls.RssAggregator.Fill 250

Ektron.Cms.Controls.RssAggregator.GetEnumerator 250

Ektron.Cms.Controls.RssAggregator.GetView 250

Ektron.Cms.Controls.RssAggregator.GetViewNames 250

Ektron.Cms.Controls.RssAggregator.IsSynchronized 252

Ektron.Cms.Controls.RssAggregator.LinkTarget 252

Ektron.Cms.Controls.RssAggregator.MaxResults 253

Ektron.Cms.Controls.RssAggregator.MoveNext 251

Ektron.Cms.Controls.RssAggregator.Reset 251

Ektron.Cms.Controls.RssAggregator.SyncRoot 253

Ektron.Cms.Controls.RssAggregator.this 251

Ektron.Cms.Controls.RssAggregator.URL 253

Ektron.Cms.Controls.RssItemBase 253

Ektron.Cms.Controls.RssItemBase.DatePosted 254

Ektron.Cms.Controls.RssItemBase.Description 254

Ektron.Cms.Controls.RssItemBase.Link 255

Ektron.Cms.Controls.RssItemBase.Title 255

Ektron.Cms.Controls.Search 255

Ektron.Cms.Controls.Search.AllowFragments 265

Ektron.Cms.Controls.Search.ButtonImgSrc 265

Ektron.Cms.Controls.Search.ButtonText 265

Ektron.Cms.Controls.Search.ContentFieldsetLegend 266

Ektron.Cms.Controls.Search.CopyTo 261

Ektron.Cms.Controls.Search.Count 266

Ektron.Cms.Controls.Search.Current 266

Ektron.Cms.Controls.Search.Display 266

Ektron.Cms.Controls.Search.EkItems 267

Ektron.Cms.Controls.Search.EmptyResultMsg 267

Ektron.Cms.Controls.Search.EnableAdvancedLink 267

Ektron.Cms.Controls.Search.EnableAjax 267

Ektron.Cms.Controls.Search.EnableArchived 268

Ektron.Cms.Controls.Search.EnableArchivedBtn 268

Ektron.Cms.Controls.Search.EnableContent 268

Ektron.Cms.Controls.Search.EnableContentBtn 269

Ektron.Cms.Controls.Search.EnableContentFieldset 269

Ektron.Cms.Controls.Search.EnableDMSAssets 269

Ektron.Cms.Controls.Search.EnableDMSAssetsBtn 269

Ektron.Cms.Controls.Search.EnableFolderFieldset 270

Ektron.Cms.Controls.Search.EnableForms 270

Ektron.Cms.Controls.Search.EnableFormsBtn 270

Ektron.Cms.Controls.Search.EnableLibFiles 271

Ektron.Cms.Controls.Search.EnableLibFilesBtn 271

Ektron.Cms.Controls.Search.EnableLibHyperlinks 271

Ektron.Cms.Controls.Search.EnableLibHyperlinksBtn 271

Ektron.Cms.Controls.Search.EnableLibImages 272

Ektron.Cms.Controls.Search.EnableLibImagesBtn 272

Ektron.Cms.Controls.Search.EnableLibraryFieldset 272

Ektron.Cms.Controls.Search.Fill 261

Ektron.Cms.Controls.Search.FolderFieldsetLegend 273

Ektron.Cms.Controls.Search.FolderID 273

Ektron.Cms.Controls.Search.FolderPath 273

Ektron.Cms.Controls.Search.GetEnumerator 261

Ektron.Cms.Controls.Search.IncludeIcons 274

Ektron.Cms.Controls.Search.IsSynchronized 274

Ektron.Cms.Controls.Search.LibraryFieldsetLegend 274

Ektron.Cms.Controls.Search.LinkTarget 275

Ektron.Cms.Controls.Search.LoadSearch 262

Ektron.Cms.Controls.Search.MaxCharacters 275

Ektron.Cms.Controls.Search.MaxResults 275
Ektron.Cms.Controls.Search.MaxTeaserLength 275
Ektron.Cms.Controls.Search.MoveNext 262
Ektron.Cms.Controls.Search.OrderBy 276
Ektron.Cms.Controls.Search.OrderDirection 276
Ektron.Cms.Controls.Search.PostResultsAt 276
Ektron.Cms.Controls.Search.Recursive 277
Ektron.Cms.Controls.Search.RemoveTeaserHtml 277
Ektron.Cms.Controls.Search.Reset 262
Ektron.Cms.Controls.Search.ResultTagId 277
Ektron.Cms.Controls.Search.Search 260
Ektron.Cms.Controls.Search.SearchDisplayUI 262
Ektron.Cms.Controls.Search.SearchType 278
Ektron.Cms.Controls.Search.ShowDate 278
Ektron.Cms.Controls.Search.ShowExtendedSearch 278
Ektron.Cms.Controls.Search.ShowSearchBoxAlways 278
Ektron.Cms.Controls.Search.ShowSearchOptions 279
Ektron.Cms.Controls.Search.Stylesheet 279
Ektron.Cms.Controls.Search.SyncRoot 279
Ektron.Cms.Controls.Search.TargetFolderDescriptionList 279
Ektron.Cms.Controls.Search.TextBoxSize 280
Ektron.Cms.Controls.Search.this 262
Ektron.Cms.Controls.Search.Wrap 263
Ektron.Cms.Controls.Search.XmlDoc 280
Ektron.Cms.Controls.Sitemap 280
Ektron.Cms.Controls.Sitemap.CacheInterval 282
Ektron.Cms.Controls.Sitemap.ClassName 282
Ektron.Cms.Controls.Sitemap.DisplayType 283
Ektron.Cms.Controls.Sitemap.Fill 282
Ektron.Cms.Controls.Sitemap.FlatTopLevel 283
Ektron.Cms.Controls.Sitemap.FolderID 284
Ektron.Cms.Controls.Sitemap.MaxLevel 284
Ektron.Cms.Controls.Sitemap.StartingLevel 284
Ektron.Cms.Controls.Sitemap.XmlDoc 285
Ektron.Cms.Controls.SmartMenu 285
Ektron.Cms.Controls.SmartMenu.AutoCollapseBranches 287
Ektron.Cms.Controls.SmartMenu.DisplayXslt 287
Ektron.Cms.Controls.SmartMenu.Enable508Compliance 287
Ektron.Cms.Controls.SmartMenu.Enable508NavigationHeadings 288
Ektron.Cms.Controls.SmartMenu.EnableMouseOverPopUp 288
Ektron.Cms.Controls.SmartMenu.EnableSmartOpen 288
Ektron.Cms.Controls.SmartMenu.EnableSubmenuLinkPathMatching 289
Ektron.Cms.Controls.SmartMenu.EnableSubmenuTemplateMatching 289
Ektron.Cms.Controls.SmartMenu.LaunchLinksInNewWindow 289
Ektron.Cms.Controls.SmartMenu.ShowRootFolder 290
Ektron.Cms.Controls.SmartMenu.SingleEditButton 290
Ektron.Cms.Controls.SmartMenu.StartCollapsed 290
Ektron.Cms.Controls.SmartMenu.StartWithRootCollapsed 291
Ektron.Cms.Controls.SmartMenu.Stylesheet 291
Ektron.Cms.Extensibility 292
Ektron.Cms.Extensibility namespace 292
Classes 292
Ektron.Cms.Extensibility.ContentInfo 293
Ektron.Cms.Extensibility.ContentInfo.AssetID 295
Ektron.Cms.Extensibility.ContentInfo.Comments 295
Ektron.Cms.Extensibility.ContentInfo.ErrorMessage 295
Ektron.Cms.Extensibility.ContentInfo.FileAndPath 295
Ektron.Cms.Extensibility.ContentInfo.FileExtension 295
Ektron.Cms.Extensibility.ContentInfo.FileName 296
Ektron.Cms.Extensibility.ContentInfo.FolderID 296
Ektron.Cms.Extensibility.ContentInfo.Html 296
Ektron.Cms.Extensibility.ContentInfo.ID 296
Ektron.Cms.Extensibility.ContentInfo.Language 296
Ektron.Cms.Extensibility.ContentInfo.MimeType 296
Ektron.Cms.Extensibility.ContentInfo.Status 297
Ektron.Cms.Extensibility.ContentInfo.Title 297
Ektron.Cms.Extensibility.ContentInfo.Type 297
Ektron.Cms.Extensibility.ContentInfo.UserID 297
Ektron.Cms.Extensibility.ExtensionAuthor 297
Ektron.Cms.Extensibility.ExtensionDescription 298
Ektron.Cms.Extensibility.ExtensionEvent 298
Ektron.Cms.Extensibility.ExtensionEvent.Content 315
Ektron.Cms.Extensibility.ExtensionEvent.Folder 316
Ektron.Cms.Extensibility.ExtensionEvent.Form 316

Ektron.Cms.Extensibility.ExtensionEvent.GetActiveDirectoryUserProxy 304
Ektron.Cms.Extensibility.ExtensionEvent.GetAssetProxy 304
Ektron.Cms.Extensibility.ExtensionEvent.GetBlogProxy 304
Ektron.Cms.Extensibility.ExtensionEvent.GetBusinessRulesProxy 305
Ektron.Cms.Extensibility.ExtensionEvent.GetCalendarEventTypeProxy 305
Ektron.Cms.Extensibility.ExtensionEvent.GetCalendarProxy 305
Ektron.Cms.Extensibility.ExtensionEvent.GetContentProxy 305
Ektron.Cms.Extensibility.ExtensionEvent.GetContentRatingProxy 306
Ektron.Cms.Extensibility.ExtensionEvent.GetCustomFieldsProxy 306
Ektron.Cms.Extensibility.ExtensionEvent.GetFolderProxy 306
Ektron.Cms.Extensibility.ExtensionEvent.GetFontProxy 306
Ektron.Cms.Extensibility.ExtensionEvent.GetFormProxy 306
Ektron.Cms.Extensibility.ExtensionEvent.GetLibraryProxy 307
Ektron.Cms.Extensibility.ExtensionEvent.GetMetadataProxy 307
Ektron.Cms.Extensibility.ExtensionEvent.GetObjectStateAsXml 307
Ektron.Cms.Extensibility.ExtensionEvent.GetPermissionsProxy 307
Ektron.Cms.Extensibility.ExtensionEvent.GetSearchManagerProxy 308
Ektron.Cms.Extensibility.ExtensionEvent.GetSiteMapProxy 308
Ektron.Cms.Extensibility.ExtensionEvent.GetSiteProxy 308
Ektron.Cms.Extensibility.ExtensionEvent.GetTaskCategoryProxy 308
Ektron.Cms.Extensibility.ExtensionEvent.GetTaskCategoryTypeProxy 308
Ektron.Cms.Extensibility.ExtensionEvent.GetTaskProxy 309
Ektron.Cms.Extensibility.ExtensionEvent.GetThreadedDiscussionProxy 309
Ektron.Cms.Extensibility.ExtensionEvent.GetUserProxy 309
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentAdd 309
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentDelete 310
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentEdit 310
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFolderAdd 310
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFolderDelete 310
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFolderEdit 311
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFormSubmit 311
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterPublish 311
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserAdd 311
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserDelete 312
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserEdit 312
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserLogin 312
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentAdd 312
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentDelete 312
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentEdit 313
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFolderAdd 313
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFolderDelete 313
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFolderEdit 313
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFormSubmit 314
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforePublish 314
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserAdd 314
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserDelete 314
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserEdit 315
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserLogin 315
Ektron.Cms.Extensibility.ExtensionEvent.Site 316
Ektron.Cms.Extensibility.ExtensionEvent.User 316
Ektron.Cms.Extensibility.FolderInfo 316
Ektron.Cms.Extensibility.FolderInfo.Description 318
Ektron.Cms.Extensibility.FolderInfo.ErrorMessage 318
Ektron.Cms.Extensibility.FolderInfo.ID 318
Ektron.Cms.Extensibility.FolderInfo.Name 318

Ektron.Cms.Extensibility.FolderInfo.ParentID 318
Ektron.Cms.Extensibility.FolderInfo.ReplicationState 319
Ektron.Cms.Extensibility.FolderInfo.StyleSheet 319
Ektron.Cms.Extensibility.FolderInfo.Type 319
Ektron.Cms.Extensibility.FormField 319
Ektron.Cms.Extensibility.FormField.FieldName 320
Ektron.Cms.Extensibility.FormField.FieldValue 320
Ektron.Cms.Extensibility.FormInfo 320
Ektron.Cms.Extensibility.FormInfo.ErrorMessage 322
Ektron.Cms.Extensibility.FormInfo.FormDescription 322
Ektron.Cms.Extensibility.FormInfo.FormFields 322
Ektron.Cms.Extensibility.FormInfo.FormTitle 323
Ektron.Cms.Extensibility.FormInfo.this 322
Ektron.Cms.Extensibility.SiteInfo 323
Ektron.Cms.Extensibility.SiteInfo.AppPath 324
Ektron.Cms.Extensibility.SiteInfo.IsSSL 324
Ektron.Cms.Extensibility.SiteInfo.Name 324
Ektron.Cms.Extensibility.SiteInfo.PhysicalSitePath 324
Ektron.Cms.Extensibility.SiteInfo.Port 325
Ektron.Cms.Extensibility.SiteInfo.Url 325
Ektron.Cms.Extensibility.UserInfo 325
Ektron.Cms.Extensibility.UserInfo.DisplayName 326
Ektron.Cms.Extensibility.UserInfo.Editor 327
Ektron.Cms.Extensibility.UserInfo.EmailAddress 327
Ektron.Cms.Extensibility.UserInfo.ErrorMessage 327
Ektron.Cms.Extensibility.UserInfo.FirstName 327
Ektron.Cms.Extensibility.UserInfo.ID 327
Ektron.Cms.Extensibility.UserInfo.Language 327
Ektron.Cms.Extensibility.UserInfo.LastName 328
Ektron.Cms.Extensibility.UserInfo.Notifications 328
Ektron.Cms.Extensibility.UserInfo.Type 328
Ektron.Cms.Extensibility.UserInfo.UserName 328
Ektron.Cms.Extensibility.WebServiceProxy 328
Ektron.Services 526
Ektron.Services namespace 526
Classes 545
Ektron.Services.ActiveDirectoryUser 545
Ektron.Services.ActiveDirectoryUser.AddADGroupToCMS 548
Ektron.Services.ActiveDirectoryUser.DeSynchUserGroups 548
Ektron.Services.ActiveDirectoryUser.DeSynchUsers 548
Ektron.Services.ActiveDirectoryUser.GetADMapping 549
Ektron.Services.ActiveDirectoryUser.GetADStatus 549
Ektron.Services.ActiveDirectoryUser.GetAvailableADGroups 550
Ektron.Services.ActiveDirectoryUser.GetAvailableADUsers 550
Ektron.Services.ActiveDirectoryUser.GetCMSGroupsToSync 551
Ektron.Services.ActiveDirectoryUser.GetCMSRelationshipsToSync 551
Ektron.Services.ActiveDirectoryUser.GetCMSUsersToSync 551
Ektron.Services.ActiveDirectoryUser.GetDomains 552
Ektron.Services.ActiveDirectoryUser.MapCMSUserGroupToAD 552
Ektron.Services.ActiveDirectoryUser.MapCMSUserToAD 553
Ektron.Services.ActiveDirectoryUser.UpdateADMapping 553
Ektron.Services.ActiveDirectoryUser.UpdateADUsersGroups 553
Ektron.Services.ActiveDirectoryUser.UpdateUsersInfoFromAD 554
Ektron.Services.Asset 554
Ektron.Services.Asset.GetAssetMgtConfigInfo 556
Ektron.Services.Asset.GetAssetMgtInfo 556
Ektron.Services.Asset.GetAssetSupertypes 557
Ektron.Services.Asset.GetCreateDragDropSnippet 557
Ektron.Services.Asset.GetCreateEktronExplorerSnippet 557
Ektron.Services.Asset.GetCreateSnippet 558
Ektron.Services.Asset.GetDragDropWOSnippet 558
Ektron.Services.Asset.GetEditSnippet 558
Ektron.Services.Asset.GetMyAssetMgtID 559
Ektron.Services.Asset.MakeCopyDMSAsset 559
Ektron.Services.Asset.SetAssetMgtConfigInfo 560
Ektron.Services.Blog 560
Ektron.Services.Blog.AddBlog 562
Ektron.Services.Blog.AddCommentForPost 563
Ektron.Services.Blog.AddPost 563
Ektron.Services.Blog.DeleteBlog 564
Ektron.Services.Blog.DeletePost 564
Ektron.Services.Blog.GetBlankBlogPostData 564
Ektron.Services.Blog.GetBlog 565, 566
Ektron.Services.Blog.GetBlogData 567, 568, 569

Ektron.Services.Blog.GetBlogDataForPost 570
Ektron.Services.Blog.GetBlogPostData 570
Ektron.Services.Blog.GetBlogRoll 571
Ektron.Services.Blog.GetPost 572
Ektron.Services.BusinessRules 572
Ektron.Services.BusinessRules.GetBusinessRules 573, 574
Ektron.Services.Calendar 575
Ektron.Services.CalendarEvent 575
Ektron.Services.CalendarEventType 575
Ektron.Services.Content 576
Ektron.Services.Content.AddContent 579, 580
Ektron.Services.Content.CheckOutContent 581
Ektron.Services.Content.CopyContentToFolder 581
Ektron.Services.Content.DeleteContentItem 582
Ektron.Services.Content.GetAddViewLanguage 582
Ektron.Services.Content.GetAllComments 583
Ektron.Services.Content.GetAllTemplates 583
Ektron.Services.Content.GetAllUnassignedItemApprovals 583
Ektron.Services.Content.GetChildContent 584
Ektron.Services.Content.GetContent 584
Ektron.Services.Content.GetContentByHistory 584
Ektron.Services.Content.GetContentForEditing 585
Ektron.Services.Content.GetContentState 585
Ektron.Services.Content.GetContentStatus 585
Ektron.Services.Content.GetDomain 586
Ektron.Services.Content.MoveContentToFolder 586
Ektron.Services.Content.PublishContent 586
Ektron.Services.Content.RenameContent 587
Ektron.Services.Content.SaveContent 587
Ektron.Services.Content.SubmitForDelete 588
Ektron.Services.Content.UndoCheckout 588
Ektron.Services.Content.UpdateContentMetaData 588
Ektron.Services.ContentRating 589
Ektron.Services.ContentRating.AddContentRating 590
Ektron.Services.ContentRating.GetContentRatingResults 590
Ektron.Services.ContentRating.PurgeContentRatings 591
Ektron.Services.CustomFields 591
Ektron.Services.CustomFields.AddNewEditableCustomFieldAssignments 594
Ektron.Services.CustomFields.BoolToYesNo 595
Ektron.Services.CustomFields.GetEditableCustomFieldAssignments 595
Ektron.Services.CustomFields.GetEnhancedMetadataArea 596
Ektron.Services.CustomFields.GetEnhancedMetadataScript 596
Ektron.Services.CustomFields.GetFieldsByFolder 597
Ektron.Services.CustomFields.GetSearchProperties 597
Ektron.Services.CustomFields.ProcessCustomFields 597
Ektron.Services.CustomFields.SearchAllAssets 598
Ektron.Services.CustomFields.WriteFilteredMetadataForEdit 598
Ektron.Services.CustomFields.WriteFilteredMetadataForView 599
Ektron.Services.CustomFields.WriteLibrarySearchExtended 599
Ektron.Services.CustomFields.WriteMetadataDefaultForEdit 599
Ektron.Services.CustomFields.WriteMetadataTypeForView 600
Ektron.Services.CustomFields.WritePublicSearch 600
Ektron.Services.Folder 601
Ektron.Services.Folder.AddFolder 602
Ektron.Services.Folder.CreateFolder 602
Ektron.Services.Folder.DeleteFolder 603
Ektron.Services.Folder.GetChildFolders 603
Ektron.Services.Folder.GetFolder 603
Ektron.Services.Folder.GetFolderID 604
Ektron.Services.Folder.GetPath 605
Ektron.Services.Folder.RenameFolder 605
Ektron.Services.Folder.UpdateFolder 605
Ektron.Services.Font 606
Ektron.Services.Font.AddFont 607
Ektron.Services.Font.DeleteFont 607
Ektron.Services.Font.GetAllFonts 607
Ektron.Services.Font.GetFont 608
Ektron.Services.Font.GetFontConfigList 608
Ektron.Services.Font.UpdateFont 608
Ektron.Services.Form 609
Ektron.Services.Form.AssignFormTask 610
Ektron.Services.Form.EmailFormFieldData 611
Ektron.Services.Form.GetAllForms 612
Ektron.Services.Form.GetForm 612
Ektron.Services.Form.GetFormDataHistogram 613

Ektron.Services.Form.GetFormFieldData 613, 614
Ektron.Services.Form.GetFormFieldList 614
Ektron.Services.Form.GetFormFieldQuestions 615
Ektron.Services.Form.GetFormTitle 615
Ektron.Services.Form.SubmitFormFieldData 616
Ektron.Services.Form.UpdateFormFieldData 616
Ektron.Services.Library 616
Ektron.Services.Library.AddLoadBalanceItem 619
Ektron.Services.Library.DeleteLibraryItem 619
Ektron.Services.Library.DeleteLibraryItemById 619
Ektron.Services.Library.DeleteLoadBalanceItem 620
Ektron.Services.Library.GetAllChildLibItems 620
Ektron.Services.Library.GetAllLoadBalancePaths 621
Ektron.Services.Library.GetLibraryItem 622
Ektron.Services.Library.GetLibrarySettings 623
Ektron.Services.Library.GetLibraryTypes 623
Ektron.Services.Library.GetLoadBalancePath 624
Ektron.Services.Library.InternalLibrarySearch 624
Ektron.Services.Library.UpdateLibrary 624
Ektron.Services.Library.UpdateLibrarySettings 625
Ektron.Services.Library.UpdateLoadBalanceSettings 625
Ektron.Services.Library.UpdateQlinkTemplates 625
Ektron.Services.Metadata 626
Ektron.Services.Metadata.AddMetaDataType 627
Ektron.Services.Metadata.DeleteMetadataType 627
Ektron.Services.Metadata.GetContentMetadataList 628
Ektron.Services.Metadata.GetMetadataType 629
Ektron.Services.Metadata.GetMetaDataTypes 629
Ektron.Services.Metadata.UpdateMetadataType 630
Ektron.Services.Permissions 630
Ektron.Services.Permissions.AddItemPermission 632
Ektron.Services.Permissions.DeleteItemPermission 632
Ektron.Services.Permissions.DisableContentInheritance 633
Ektron.Services.Permissions.DisableFolderInheritance 633
Ektron.Services.Permissions.EnableContentInheritance 633
Ektron.Services.Permissions.EnableFolderInheritance 634
Ektron.Services.Permissions.GetUserPermissions 635
Ektron.Services.Permissions.LoadPermissions 636
Ektron.Services.Permissions.UpdateItemPermission 636
Ektron.Services.SearchManager 636
Ektron.Services.SearchManager.ExecuteSearch 638
Ektron.Services.SearchManager.Search 639, 640
Ektron.Services.Site 640
Ektron.Services.Site.BreakTaskPermissionInherit 642
Ektron.Services.Site.GetAllActiveLanguages 643
Ektron.Services.Site.GetAllLanguages 643
Ektron.Services.Site.GetClientDayNames 643
Ektron.Services.Site.GetClientMonthNames 644
Ektron.Services.Site.GetCookieValue 644
Ektron.Services.Site.GetLanguage 644
Ektron.Services.Site.GetLanguages 645
Ektron.Services.Site.GetSiteVariables 645
Ektron.Services.Site.GetTaskPermission 645
Ektron.Services.Site.IsTaskPermissionInherited 646
Ektron.Services.Site.SetCookieValue 646
Ektron.Services.Site.SetTaskPermission 647
Ektron.Services.Site.UpdateSiteEnabledLanguages 647
Ektron.Services.SiteMap 647
Ektron.Services.SiteMap.GetFolderBreadcrumbPath 649
Ektron.Services.SiteMap.GetWebsiteMap 649
Ektron.Services.SiteMap.RenderFolderBreadcrumbHtml 649
Ektron.Services.SiteMap.RenderWebsiteMap 650
Ektron.Services.Task 526
Ektron.Services.Task namespace 526
Classes 527
Ektron.Services.Task.Task 527
Ektron.Services.Task.Task.AddTask 529
Ektron.Services.Task.Task.AssignTaskToUser 529, 530
Ektron.Services.Task.Task.AssignTaskToUsergroup 530, 531
Ektron.Services.Task.Task.DeleteTask 531, 532
Ektron.Services.Task.Task.DeleteTasks 532, 533
Ektron.Services.Task.Task.GetTask 533
Ektron.Services.Task.Task.GetTasks 534
Ektron.Services.Task.Task.GetTasksAssignedByUser 534, 535
Ektron.Services.Task.Task.GetTasksAssignedToUser 535, 536
Ektron.Services.Task.Task.GetTasksAssignedToUsergroup 536, 537
Ektron.Services.Task.Task.GetTaskState 537
Ektron.Services.Task.Task.SetStateForTask 538

Ektron.Services.Task.Task.UpdateTask 538
Ektron.Services.Task.TaskCategory 538
Ektron.Services.Task.TaskCategory.AddTaskCategory 540
Ektron.Services.Task.TaskCategory.DeleteTaskCategory 540
Ektron.Services.Task.TaskCategory.GetAllCategories 540
Ektron.Services.Task.TaskCategory.GetTaskCategory 541
Ektron.Services.Task.TaskCategory.UpdateTaskCategory 541
Ektron.Services.Task.TaskCategoryType 542
Ektron.Services.Task.TaskCategoryType.AddTaskType 543
Ektron.Services.Task.TaskCategoryType.DeleteTaskType 543
Ektron.Services.Task.TaskCategoryType.GetTaskType 544
Ektron.Services.Task.TaskCategoryType.UpdateTaskType 544
Ektron.Services.ThreadedDiscussion 650
Ektron.Services.ThreadedDiscussion.AddBoard 653
Ektron.Services.ThreadedDiscussion.AddForum 653
Ektron.Services.ThreadedDiscussion.AddTopic 654
Ektron.Services.ThreadedDiscussion.DeleteBoard 655
Ektron.Services.ThreadedDiscussion.DeleteForum 655
Ektron.Services.ThreadedDiscussion.DeleteTopic 655
Ektron.Services.ThreadedDiscussion.GetBoardCategories 656
Ektron.Services.ThreadedDiscussion.GetBoardForums 656
Ektron.Services.ThreadedDiscussion.GetDiscussionBoard 657
Ektron.Services.ThreadedDiscussion.GetDiscussionBoardString 658
Ektron.Services.ThreadedDiscussion.GetForum 658
Ektron.Services.ThreadedDiscussion.GetRepliesForTopics 658
Ektron.Services.ThreadedDiscussion.GetTopic 659
Ektron.Services.ThreadedDiscussion.SearchReplies 659
Ektron.Services.User 660
Ektron.Services.User.ActivateUserAccount 665, 666
Ektron.Services.User.AddMembershipUser 666
Ektron.Services.User.AddNewUser 666
Ektron.Services.User.AddUser 667
Ektron.Services.User.AddUserGroup 667
Ektron.Services.User.AddUserToGroup 668
Ektron.Services.User.AutoLogInUser 668
Ektron.Services.User.CreateBlankUserObject 669
Ektron.Services.User.DeleteUserByID 669
Ektron.Services.User.DeleteUserByIds 670
Ektron.Services.User.DeleteUserFromGroup 670
Ektron.Services.User.DeleteUserGroup 671
Ektron.Services.User.EditUserCustomProperties 671
Ektron.Services.User.GetActiveUser 671
Ektron.Services.User.GetADMapping 672
Ektron.Services.User.GetAllCmsUsers 672
Ektron.Services.User.GetAllCustomProperties 673
Ektron.Services.User.GetAllUserByCustomProperty 673
Ektron.Services.User.GetAllUserGroups 674
Ektron.Services.User.GetAllUsers 675, 676
Ektron.Services.User.GetGroupsUserIsIn 676
Ektron.Services.User.GetMembershipUser 676
Ektron.Services.User.GetUser 677
Ektron.Services.User.GetUserbyUsername 677
Ektron.Services.User.GetUserCustomProperties 678
Ektron.Services.User.GetUserEmailInfo 678
Ektron.Services.User.GetUserForFolderAdmin 679
Ektron.Services.User.GetUserGroup 679
Ektron.Services.User.GetUserGroupForFolderAdmin 679
Ektron.Services.User.GetUserPermissions 680
Ektron.Services.User.GetUserPreference 680
Ektron.Services.User.GetUsers 681, 682
Ektron.Services.User.GetUsersForTask 682
Ektron.Services.User.GetUsersNotInGroup 683
Ektron.Services.User.IsARoleMember 684, 685
Ektron.Services.User.IsARoleMemberForContent 686
Ektron.Services.User.IsARoleMemberForFolder 686
Ektron.Services.User.LogInUser 687
Ektron.Services.User.ResetMembershipUserPassword 687
Ektron.Services.User.UnsubscribeUser 687
Ektron.Services.User.UpdateCustomPropertiesItemOrder 688
Ektron.Services.User.UpdateUser 689
Ektron.Services.User.UpDateUserGroup 689
Ektron.Services.User.UpdateUserPreferences 690
Ektron.Services.User.UserObject 690, 691
EktronCatalogPart class 114
about EktronCatalogPart class 114
DefaultContentID 115
EktronCatalogPart properties 114

EktronCommunityCatalogPart class 115
about EktronCommunityCatalogPart class 115
EktronWebPartZone class 115
about EktronWebPartZone class 115
CmsSitePath 116
EktronWebPartZone properties 116
EkWebControl class 117
about EkWebControl class 117
Authenticated 122
EkWebControl 119
EkWebControl methods 119
EkWebControl properties 121
Error 122
Fill 119
Hide 122
IsLoggedIn 123
Language 123
LanguageID 123
LocalizeString 120
loggedInUserID 123
loggedInUserName 124
RegisterClientScriptInclude 120
RegisterStylesheet 120
SeachHeader 120
SuppressWrapperTags 124
Text 124
ToString 121
Wrap 121
WrapTag 124
EkWebPart class 125
about EkWebPart class 125
EkXsltBase class 125
about EkXsltBase class 125
DisplayXslt 127
EkXsltBase 126
EkXsltBase methods 127
EkXsltBase properties 127
SetDirty 127
XmlDoc 128
XsltDoc 128
EkXsltWebPart class 128
about EkXsltWebPart class 128
ExtensionAuthor class 297
about ExtensionAuthor class 297
ExtensionDescription class 298
about ExtensionDescription class 298
ExtensionEvent class 298
about ExtensionEvent class 298
Content 315
ExtensionEvent methods 302
ExtensionEvent properties 315
Folder 316
Form 316
GetActiveDirectoryUserProxy 304
GetAssetProxy 304
GetBlogProxy 304
GetBusinessRulesProxy 305
GetCalendarEventTypeProxy 305
GetCalendarProxy 305
GetContentProxy 305
GetContentRatingProxy 306
GetCustomFieldsProxy 306
GetFolderProxy 306
GetFontProxy 306
GetFormProxy 306
GetLibraryProxy 307
GetMetadataProxy 307
GetObjectStateAsXml 307
GetPermissionsProxy 307
GetSearchManagerProxy 308
GetSiteMapProxy 308
GetSiteProxy 308
GetTaskCategoryProxy 308
GetTaskCategoryTypeProxy 308
GetTaskProxy 309
GetThreadedDiscussionProxy 309
GetUserProxy 309
OnAfterContentAdd 309

OnAfterContentDelete 310
OnAfterContentEdit 310
OnAfterFolderAdd 310
OnAfterFolderDelete 310
OnAfterFolderEdit 311
OnAfterFormSubmit 311
OnAfterPublish 311
OnAfterUserAdd 311
OnAfterUserDelete 312
OnAfterUserEdit 312
OnAfterUserLogin 312
OnBeforeContentAdd 312
OnBeforeContentDelete 312
OnBeforeContentEdit 313
OnBeforeFolderAdd 313
OnBeforeFolderDelete 313
OnBeforeFolderEdit 313
OnBeforeFormSubmit 314
OnBeforePublish 314
OnBeforeUserAdd 314
OnBeforeUserDelete 314
OnBeforeUserEdit 315
OnBeforeUserLogin 315
Site 316
User 316

F

Folder class 466, 601
about Folder class 466, 601
AddFolder 468, 602
CreateFolder 468, 602
DeleteFolder 603
DeleteFolderById 469
Folder methods 468, 602
GetChildFolders 469, 603
GetFolder 469, 603
GetFolderDataWithPermission 470
GetFolderId 470, 471
GetFolderID 604
GetFolderId method 470
GetFolderID method 604
GetFolderInfo 471
GetPath 471, 605
RenameFolder 472, 605
UpdateFolder 472, 605
FolderBreadcrumb class 128
about FolderBreadcrumb class 128
DynamicParameter 130
Fill 130
FolderBreadcrumb methods 129
FolderBreadcrumb properties 130
FolderInfo class 316
about FolderInfo class 316
Description 318
ErrorMessage 318
FolderInfo properties 317
ID 318
Name 318
ParentID 318
ReplicationState 319
StyleSheet 319
Type 319
Font class 472, 606
about Font class 472, 606
AddFont 473, 607
DeleteFont 474, 607
Font methods 473, 606
GetAllFonts 474, 607
GetFont 474, 608
GetFontConfigList 474, 608
UpdateFont 475, 608
Form class 383, 609
about Form class 383, 609
AssignFormTask 385, 610
EmailFormFieldData 385, 611
Form methods 384, 610
GetAllForms 386, 612
GetAllForms method 386, 611

GetForm 387, 612
GetFormDataHistogram 387, 613
GetFormFieldData 387, 388, 613, 614
GetFormFieldData method 387, 613
GetFormFieldList 388, 614
GetFormFieldQuestions 389, 615
GetFormTitle 389, 615
SubmitFormFieldData 389, 616
UpdateFormFieldData 390, 616
FormBlock class 130
about FormBlock class 130
AddValidation 136
CacheInterval 136
CopyTo 133
Count 136
Current 136
DynamicParameter 137
EkItem 137
Fields 137
Fill 133
FormBlock methods 132
FormBlock properties 135
FormName 137
GetEnumerator 133
GetView 133
GetViewNames 134
IncludeTags 138
IsSynchronized 138
MoveNext 134
Reset 134
StartingFolderID 138
SyncRoot 139
this 134
ToString 135
Wrap 135
FormField class 319
about FormField class 319
FieldName 320
FieldValue 320
FormField properties 320
FormFieldItem class 139
about FormFieldItem class 139
Basetype 140
ContentType 140
DatalistItems 141
Datatype 141
DisplayName 141
FieldName 142
FormFieldItem properties 140
Language 142
Value 142
FormFieldItemDatalistItem class 142
about FormFieldItemDatalistItem class 142
DataValue 143
DisplayValue 144
FormFieldItemDatalistItem properties 143
FormInfo class 320
about FormInfo class 320
ErrorMessage 322
FormDescription 322
FormFields 322
FormInfo methods 321
FormInfo properties 322
FormTitle 323
this 322
Forum class 144
about Forum class 144
CacheInterval 146
Fill 145
Forum methods 145
Forum properties 146
ToString 145
Wrap 146

H

HtmlEditDisplay class 147
about HtmlEditDisplay class 147
BaseJavaScript 148

BeginJavaScript 149
EndJavaScript 149
HtmlEditDisplay methods 148
MakeFieldNameFromId 149
RenderCMSFields 150
SelectedJavaScript 150
HtmlEditor class 150
about HtmlEditor class 150
Archived 157
Authenticated 157
CmsSitePath 158
ContentAction 158
ContentTitle 159
ContentType 159
EditorReadyCall 160
EnableUpload 160
ExecuteContentAction 155
HtmlEditor 154
HtmlEditor methods 154
HtmlEditor properties 156
IncludeContentInfo 161
IsLoggedIn 161
loggedInUserID 162
loggedInUserName 162
Path 162
PreferredType 163
SafeEncode 163
ShowCMSToolbar 164
ShowTitle 164
StyleSheet 165
Text 165
ToolbarLevel 166
ToolbarResetCall 166
ToString 155
TranslationFile 167
WorkareaMode 155

IndexSearch class 167
about IndexSearch class 167
AddParm 173
AddParm method 173
ButtonImgSrc 177
ButtonText 178
ContentParameter 178
CopyTo 173
Count 178
Current 178
EkItems 179
EmptyResultMsg 179
EncodeContentHtml 179
Fill 173
FolderId 179
FormName 180
GetEnumerator 174
GetSearchParmDataType 174
GetSearchParmSearchType 174
GetXPathValues 174
HitsMarker 180
IndexSearch 171
IndexSearch methods 172
IndexSearch properties 176
IndexSearch structures 171
IsSynchronized 180
Labelafter 180
Labelbefore 181
LabelBetween 181
LabelContains 181
LabeldateBetween 182
LabelEqual 182
LabelExactPhrase 182
LabelGreaterThan 183
Labelinputdate 183
Labelinputnumber 183
LabelLessThan 184
Labelon 184
LabelSelect 184
LinkTarget 185

MaxResults 185
MoveNext 175
OrderBy 185
OrderByDirection 185
Recursive 186
Reset 175
Search 175
Search method 175
SearchParmXML 186
ShowSearchBoxAlways 186
SyncRoot 186
this 176
Weighted 187
Wrap 176
XmlConfigId 187
XmlDoc 187

L

LanguageAPI class 187
about LanguageAPI class 187
CurrentLanguage 191
CurrentLanguageID 192
DefaultLanguage 192
DefaultLanguageID 192
Fill 190
GetLanguage 190
GetLanguageID 190
IsValid 191
IsValid method 190
LangaugeIdList 192
LangaugeTitleList 192
LanguageAPI 189
LanguageAPI methods 190
LanguageAPI properties 191
MultiLanguageEnabled 193
OverrideDefault 193
SiteLanguage 193
SiteLanguageID 193
LanguageSelect class 194
about LanguageSelect class 194
Fill 195
LabelName 196
LanguageSelect methods 195
LanguageSelect properties 195
Text 196
Wrap 195
Library class 475, 616
about Library class 475, 616
AddLibraryItem 477
AddLoadBalanceItem 478, 619
DeleteLibraryItem 478, 619
DeleteLibraryItemById 478, 619
DeleteLoadBalanceItem 479, 620
GetAllChildLibItems 479, 620
GetAllLoadBalancePaths 480, 621
GetAllLoadBalancePaths method 479, 621
GetLibraryItem 481, 622
GetLibraryItem method 480, 622
GetLibrarySettings 481, 623
GetLibraryTypes 482, 623
GetLoadBalancePath 482, 624
InternalLibrarySearch 482, 624
Library methods 477, 618
UpdateLibrary 483, 624
UpdateLibrarySettings 483, 625
UpdateLoadBalanceSettings 483, 625
UpdateQlinkTemplates 484, 625
List of Developer Resources 2
ListSummary class 196
about ListSummary class 196
CacheInterval 202
ContentParameter 203
ContentType 203
CopyTo 199
Count 203
Current 203
Direction 204
EkItems 204

EnablePaging 204
Fill 199
FolderID 205
FolderPath 205
GetEnumerator 199
GetHtml 205
GetRssFeed 200
GetView 200
GetViewNames 200
IncludeIcons 206
IsSynchronized 206
LinkTarget 206
ListSummary methods 199
ListSummary properties 202
MaxResults 207
MoveNext 200
OrderKey 207
Random 207
Recursive 208
Reset 201
SyncRoot 208
this 201
ToString 201
Wrap 201
XmlDoc 208
Localization class 484
about Localization class 484
GetAddViewLanguage 487
GetAddViewLanguageForAllContent 487
GetAddViewLanguageForMenus 488
GetContentItems 488
GetFile 488
GetFiles 489
GetFilesInZipFile 489
GetFlagUrl 490
GetFlagUrl method 489
GetInterchangeFile 491
GetInterchangeFiles 491
GetInterchangeFilesInZipFile 492
GetJobs 492, 493
GetJobs method 492
GetLocalizationUrl 493
GetTranslationUploadDirectory 493
Localization methods 486
StartExportContentForTranslation 493
StartExportFolderForTranslation 494
StartExportMenusForTranslation 494
StartImportTranslation 495
Login class 208
about Login class 208
AutoAddType 210
AutoLogin 211
Fill 210
Login methods 210
Login properties 210
OnlyAllowMemberLogin 211
Password 211
PromptLogout 211
SuppressHelpButton 211
Text 212
UserName 212

M

Membership class 212
about Membership class 212
BoardID 215
DisplayMode 216
Fill 215
IsPostBack 214
Membership fields 214
Membership methods 214
Membership properties 215
RedirectFailedURL 216
RedirectSuccessURL 216
RegisterButtonImg 216
RegisterButtonText 217
ResetButtonImg 217
ResetButtonText 217

ShowExtended 217
ShowTerms 218
Text 218
UserExistsMessage 218
UserID 218
UserSuccessMessage 218
UserUpdateSuccessMessage 219
Menu class 219
about Menu class 219
CacheInterval 221
DefaultMenuID 222
DisplayXslt 222
DynamicParameter 222
Fill 221
Menu methods 220
Menu properties 221
Stylesheet 223
Wrap 221
Metadata class 495, 626
about Metadata class 495, 626
AddMetaDataType 497, 627
AddMetaDataType method 496
DeleteMetadataType 497, 627
GetContentMetadataList 498, 628
GetContentMetadataList method 498, 628
GetMetadataType 499, 629
GetMetaDataTypes 499, 629
Metadata methods 496, 627
UpdateMetadataType 499, 630
MetaData class 223
about MetaData class 223
DefaultContentID 224
Fill 224
MetaData methods 224
MetaData properties 224
MetaDataList class 225
about MetaDataList class 225
CacheInterval 230
ContentType 230
CopyTo 228
Count 230
Current 230
EkItems 231
EnablePaging 231
ExactPhrase 231
Fill 228
FolderID 232
GetEnumerator 228
GetHtml 232
IncludeIcons 232
IsSynchronized 233
KeyWordName 233
KeyWordValue 233
LinkTarget 233
MaxNumber 234
MetaDataList methods 227
MetaDataList properties 229
MoveNext 228
OrderBy 234
Recursive 234
Reset 228
SortOrder 235
SyncRoot 235
this 229
Wrap 229
XmlDoc 235

P

Permissions class 500, 630
about Permissions class 500, 630
AddItemPermission 501, 632
DeleteItemPermission 502, 632
DisableContentInheritance 502, 633
DisableFolderInheritance 502, 633
EnableContentInheritance 503, 633
EnableFolderInheritance 503, 634
GetUserPermissions 503, 504, 635
GetUserPermissions method 503, 634

LoadPermissions 504, 636
Permissions methods 501, 631
UpdateItemPermission 505, 636
PersonalizationManager class 235
about PersonalizationManager class 235
AddContentExitText 238
AddContentText 238
CurrentUserButtonText 238
Display 239
Fill 237
HeaderBackColor 239
HeaderText 239
PersonalizationManager methods 237
PersonalizationManager properties 237
PersonalizeExitText 240
PersonalizeScopeText 240
PersonalizeStartText 240
PublicUserButtonText 241
ResetLinkText 241
ResetLinkToolTip 242
Plug-in Extension API 291
Poll class 242
about Poll class 242
EnableAjax 244
Fill 243
Poll methods 243
Poll properties 244
PostHistory class 244
about PostHistory class 244
BoardID 247
CacheInterval 247
Fill 246
PostHistory methods 245
PostHistory properties 246
ToString 246
Wrap 246

R

RssAggregator class 247
about RssAggregator class 247
CopyTo 249
Count 251
Current 252
EkItems 252
Fill 250
GetEnumerator 250
GetView 250
GetViewNames 250
IsSynchronized 252
LinkTarget 252
MaxResults 253
MoveNext 251
Reset 251
RssAggregator methods 249
RssAggregator properties 251
SyncRoot 253
this 251
URL 253
RssItemBase class 253
about RssItemBase class 253
DatePosted 254
Description 254
Link 255
RssItemBase fields 254
Title 255

S

Search class 255
about Search class 255
AllowFragments 265
ButtonImgSrc 265
ButtonText 265
ContentFieldsetLegend 266
CopyTo 261
Count 266
Current 266
Display 266
EkItems 267

EmptyResultMsg 267
EnableAdvancedLink 267
EnableAjax 267
EnableArchived 268
EnableArchivedBtn 268
EnableContent 268
EnableContentBtn 269
EnableContentFieldset 269
EnableDMSAssets 269
EnableDMSAssetsBtn 269
EnableFolderFieldset 270
EnableForms 270
EnableFormsBtn 270
EnableLibFiles 271
EnableLibFilesBtn 271
EnableLibHyperlinks 271
EnableLibHyperlinksBtn 271
EnableLibImages 272
EnableLibImagesBtn 272
EnableLibraryFieldset 272
Fill 261
FolderFieldsetLegend 273
FolderID 273
FolderPath 273
GetEnumerator 261
IncludeIcons 274
IsSynchronized 274
LibraryFieldsetLegend 274
LinkTarget 275
LoadSearch 262
MaxCharacters 275
MaxResults 275
MaxTeaserLength 275
MoveNext 262
OrderBy 276
OrderDirection 276
PostResultsAt 276
Recursive 277
RemoveTeaserHtml 277
Reset 262
ResultTagId 277
Search 260
Search methods 261
Search properties 263
SearchDisplayUI 262
SearchType 278
ShowDate 278
ShowExtendedSearch 278
ShowSearchBoxAlways 278
ShowSearchOptions 279
Stylesheet 279
SyncRoot 279
TargetFolderDescriptionList 279
TextBoxSize 280
this 262
Wrap 263
XmlDoc 280
SearchManager class 400, 636
about SearchManager class 400, 636
Execute 401, 402
Execute method 401
ExecuteSearch 638
ExecuteSearch method 637
Search 639, 640
Search method 639
SearchManager methods 400, 637
Server Controls API 5
Site class 505, 640
about Site class 505, 640
BreakTaskPermissionInherit 507, 642
DeleteCookie 508
GetAllActiveLanguages 508, 643
GetAllLanguages 508, 643
GetClientDayNames 508, 643
GetClientMonthNames 509, 644
GetCookieValue 509, 644
GetLanguage 509, 644
GetLanguages 510, 645

GetSiteVariables 510, 645
GetTaskPermission 510, 645
IsTaskPermissionInherited 511, 646
SetCookieValue 511, 646
SetTaskPermission 511, 647
Site methods 507, 642
UpdateSiteEnabledLanguages 512, 647
UpdateSiteVariables 512
SiteInfo class 323
about SiteInfo class 323
AppPath 324
IsSSL 324
Name 324
PhysicalSitePath 324
Port 325
SiteInfo properties 324
Url 325
Sitemap class 280
about Sitemap class 280
CacheInterval 282
ClassName 282
DisplayType 283
Fill 282
FlatTopLevel 283
FolderID 284
MaxLevel 284
Sitemap methods 281
Sitemap properties 282
StartingLevel 284
XmlDoc 285
SiteMap class 512, 647
about SiteMap class 512, 647
GetFolderBreadcrumbPath 513, 649
GetWebsiteMap 514, 649
RenderFolderBreadcrumbHtml 514, 649
RenderWebsiteMap 514, 650
SiteMap methods 513, 648
SmartMenu class 285
about SmartMenu class 285
AutoCollapseBranches 287
DisplayXslt 287
Enable508Compliance 287
Enable508NavigationHeadings 288
EnableMouseOverPopUp 288
EnableSmartOpen 288
EnableSubmenuLinkPathMatching 289
EnableSubmenuTemplateMatching 289
LaunchLinksInNewWindow 289
ShowRootFolder 290
SingleEditButton 290
SmartMenu properties 286
StartCollapsed 290
StartWithRootCollapsed 291
Stylesheet 291
Subscription class 515
about Subscription class 515
AddEmailFrom 518
AddSubscription 518
AddSubscriptionMessage 518
DeleteEmailFrom 519
DeleteSubscription 519
DeleteSubscriptionMessage 519
GetAllActiveSubscriptions 519
GetAllEmailFrom 520
GetAllSubscriptionMessages 520
GetAllSubscriptions 520
GetFolderInheritedFrom 521
GetSubscription 521
GetSubscriptionMessage 521
GetSubscriptionMessagesForType 521
GetSubscriptionPropertiesForContent 522
GetSubscriptionPropertiesForFolder 522
GetSubscriptionsForContent 522
GetSubscriptionsForFolder 523
Subscription methods 517
UpdateEmailFrom 523
UpdateSubscription 523
UpdateSubscriptionMessage 524

T

Task class 527
- about Task class 527
- AddTask 529
- AssignTaskToUser 529, 530
- AssignTaskToUser method 529
- AssignTaskToUsergroup 530, 531
- AssignTaskToUsergroup method 530
- DeleteTask 531, 532
- DeleteTask method 531
- DeleteTasks 532, 533
- DeleteTasks method 532
- GetTask 533
- GetTasks 534
- GetTasksAssignedByUser 534, 535
- GetTasksAssignedByUser method 534
- GetTasksAssignedToUser 535, 536
- GetTasksAssignedToUser method 535
- GetTasksAssignedToUsergroup 536, 537
- GetTasksAssignedToUsergroup method 536
- GetTaskState 537
- SetStateForTask 538
- Task methods 528
- UpdateTask 538

TaskCategory class 538
- about TaskCategory class 538
- AddTaskCategory 540
- DeleteTaskCategory 540
- GetAllCategories 540
- GetTaskCategory 541
- TaskCategory methods 539
- UpdateTaskCategory 541

TaskCategoryType class 542
- about TaskCategoryType class 542
- AddTaskType 543
- DeleteTaskType 543
- GetTaskType 544
- TaskCategoryType methods 543
- UpdateTaskType 544

ThreadedDiscussion class 390, 650
- about ThreadedDiscussion class 390, 650
- AddBoard 393, 653
- AddForum 393, 653
- AddTopic 394, 654
- AddTopic method 394, 654
- DeleteBoard 395, 655
- DeleteForum 395, 655
- DeleteTopic 395, 655
- GetBoardCategories 396, 656
- GetBoardForums 396, 656
- GetDiscussionBoard 397, 657
- GetDiscussionBoard method 396, 657
- GetDiscussionBoardString 397, 658
- GetForum 398, 658
- GetRepliesForTopics 398, 658
- GetTopic 398, 659
- SearchReplies 399, 659
- ThreadedDiscussion methods 392, 652

U

User class 413, 660
- about User class 413, 660
- ActivateUserAccount 419, 665, 666
- ActivateUserAccount method 418, 665
- AddMembershipUser 419, 666
- AddNewUser 420, 666
- AddRolePermission 420
- AddUser 421, 667
- AddUser method 420
- AddUserGroup 421, 667
- AddUserToGroup 421, 668
- AutoLogInUser 422, 668
- CreateBlankUserObject 422, 423, 669
- CreateBlankUserObject method 422, 668
- DeleteUserByID 423, 669
- DeleteUserByIds 423, 670
- DeleteUserFromGroup 423, 670

DeleteUserGroup 424, 671
EditUserCustomProperties 424, 671
GetActiveUser 424, 671
GetADMapping 424, 672
GetAllCmsUsers 425, 672
GetAllCustomProperties 425, 673
GetAllUserByCustomProperty 426, 673
GetAllUserGroups 426, 427, 674
GetAllUserGroups method 426, 674
GetAllUsers 427, 428, 675, 676
GetAllUsers method 427, 675
GetGroupsUserIsIn 428, 676
GetMembershipUser 428, 676
GetUser 428, 677
GetUserbyUsername 677
GetUserByUsername 429
GetUserCustomProperties 429, 678
GetUserEmailInfo 429, 678
GetUserForFolderAdmin 430, 679
GetUserGroup 430, 679
GetUserGroupByName 430
GetUserGroupForFolderAdmin 431, 679
GetUserPermissions 431, 680
GetUserPreference 431, 680
GetUsers 432, 433, 681, 682
GetUsers method 432, 681
GetUsersForTask 433, 682
GetUsersNotInGroup 433, 683
IsAdmin 434
IsARoleMember 435, 436, 684, 685
IsARoleMember method 434, 683
IsARoleMemberForContent 436, 686
IsARoleMemberForContent_FolderUserAdmin 437
IsARoleMemberForFolder 437, 686
IsARoleMemberForFolder_FolderUserAdmin 438
IsARoleMemberForFolder_MetadataAdmin 438
LogInUser 438, 687
ReadCustomProperties 439
ResetMembershipUserPassword 439, 687
SetAuthenticationCookie 439
UnsubscribeUser 440, 687
UpdateCustomPropertiesItemOrder 440, 688
UpdateUser 441, 689
UpdateUser method 441, 688
UpDateUserGroup 441, 442, 689
UpDateUserGroup method 441
UpdateUserPreferences 442, 690
User methods 416, 663
UserObject 442, 443, 690, 691
UserObject method 442, 690
UserInfo class 325
about UserInfo class 325
DisplayName 326
Editor 327
EmailAddress 327
ErrorMessage 327
FirstName 327
ID 327
Language 327
LastName 328
Notifications 328
Type 328
UserInfo properties 326
UserName 328
Utilities class 524
about Utilities class 524
LoadPermissions 525
Utilities methods 525

W

Web Services API 525
WebServiceProxy class 328
about WebServiceProxy class 328
Welcome 1